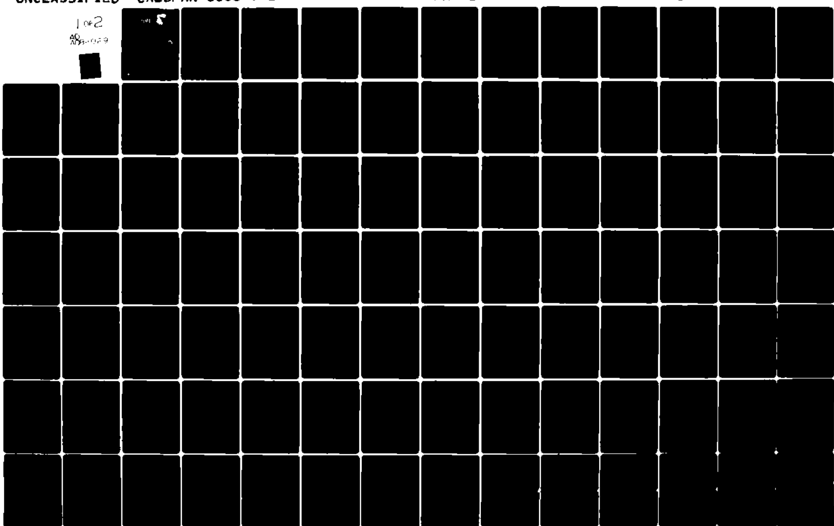
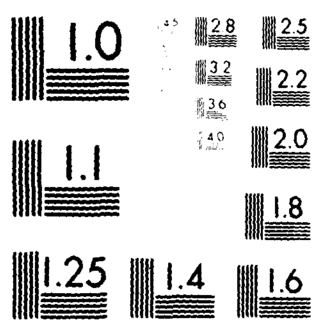
AD-A088 029 CALSPAN ADVANCED TECHNOLOGY CENTER BUFFALO NY F/G 1/2
ADVANCED RESTRAINT SYSTEM MODELING.(U)
MAY 80 F E BUTLER, J T FLECK F33615-78-C-0516
UNCLASSIFIED CALSPAN-6306-V-1 AFAMRL-TR-80-14 NL

1 OF 2
AD
A088029





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS

AFAMRL-TR-80-14

LEVEL II

ADVANCED RESTRAINT SYSTEM MODELING

F. E. BUTLER
CALSPAN CORPORATION
ADVANCED TECHNOLOGY CENTER
BUFFALO, NEW YORK 14225

J. T. FLECK
J & J TECHNOLOGIES INC.
ORCHARD PARK, NEW YORK 14127

MAY 1980

AIR FORCE AEROSPACE MEDICAL RESEARCH LABORATORY
AEROSPACE MEDICAL DIVISION
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433

NOTICES

When US Government drawings, specifications, or other data are used for any purpose other than a definitely related Government procurement operation, the Government thereby incurs no responsibility nor any obligation whatsoever, and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise, as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

Please do not request copies of this report from Air Force Aerospace Medical Research Laboratory. Additional copies may be purchased from:

National Technical Information Service
5285 Port Royal Road
Springfield, Virginia 22161

Federal Government agencies and their contractors registered with Defense Documentation Center should direct requests for copies of this report to:

Defense Documentation Center
Cameron Station
Alexandria, Virginia 22314

TECHNICAL REVIEW AND APPROVAL

AFAMRL-TR-80-14

This report has been reviewed by the Office of Public Affairs (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.

FOR THE COMMANDER

HENNING E. VON GIERKE
Director
Biodynamics and Bioengineering Division
Air Force Aerospace Medical Research Laboratory

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: (18) AFAMRL-TR-80-14 (19)

2. GOVT ACCESSION NO.: AD-A088 029

3. RECIPIENT'S CATALOG NUMBER:

4. TITLE (and Subtitle): (6) ADVANCED RESTRAINT SYSTEM MODELING (14)

5. TYPE OF REPORT & PERIOD COVERED: (9) Final Technical Report, April 1978 – November 1979

6. PERFORMING ORG. REPORT NUMBER: CALSPAN-6306-V-1

7. AUTHOR(s): (10) Frank E. Butler, John T. Fleck

8. CONTRACT OR GRANT NUMBER(s): (15) F33615-78-C-0516

9. PERFORMING ORGANIZATION NAME AND ADDRESS:
Calspan ~~Corporation~~ Advanced Technology Center
P.O. Box 400
Buffalo, NY 14225

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: (17) 62202F; 7231-15-07 (16)

11. CONTROLLING OFFICE NAME AND ADDRESS:
Air Force Aerospace Medical Research Laboratory,
AMD, AFSC,
Wright-Patterson Air Force Base, Ohio 45433

12. REPORT DATE: (11) May 1980

13. NUMBER OF PAGES: 176

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): (12) 176

15. SECURITY CLASS. (of this report): UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE:

16. DISTRIBUTION STATEMENT (of this Report):

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report):

18. SUPPLEMENTARY NOTES:

Previous development of the Calspan 3-Dimensional Crash Victim Sinulation (CVS) program was sponsored by the National Highway Traffic Safety Administration, the Motor Vehicle Manufacturers Association and the AF Aerospace Medical Research Laboratory.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number):

COMPUTER SIMULATION
THREE DIMENSION DYNAMICS
MATHEMATICAL MODEL
CRASH VICTIM SIMULATION
HARNESS RESTRAINT SYSTEMS
AERODYNAMIC FORCES

20. ABSTRACT (Continue on reverse side if necessary and identify by block number):

The Articulated Total Body (ATB) model is currently being used by the AFAMRL to study the biomechanics of the pilot-seat ejection from an aircraft. The new ATB-II model presented in this report incorporates features developed since the original ATB model completion and new mathematical algorithms designed to improve the usefulness and correct some of the deficiencies of the model. The new features developed for this research program are a new harness-belt system, rate dependent force producing functions, arbitrary specification

(Continued on reverse)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

410863

7/

20. ABSTRACT (cont)

of the motion of multiple segments and the computation of the initial orientation from orthogonal projections of the segment axes.

PREFACE

This report describes the combination of the Articulated Total Body (ATB) model and the Calspan 3-D Crash Victim Simulation (CVS) program and modifications made to the program to form the new ATB-II model.

The principal modifications described herein fall into four categories as follows:

- New harness-belt algorithm
- Rate dependent functions
- Arbitrary specification of the motion of multiple segments
- Projection of segment axes to specify initial angular orientation

The research effort summarized in this report was performed for the Air Force Aerospace Medical Research Laboratory under Contract No. F33615-78-C-0516. Frank ~~E.~~ Butler of the Transportation Research Department, Advanced Technology Center, Calspan Corporation served as project engineer and developed the computer software required. Dr. ~~John T.~~ Fleck of J&J Technologies Inc., a consultant to Calspan, developed the mathematical algorithms.

The authors wish to acknowledge Ints Kaleps of the Air Force Aerospace Medical Research Laboratory for his suggestions and guidance during the analytical and software development of this research effort.

TABLE OF CONTENTS

LIST OF FIGURES

INTRODUCTION

HISTORY OF THE ATB MODEL AND ITS RELATIONSHIP TO THE CVS PROGRAM

The original Articulated Total Body (ATB) model was an AMRL modification of version 12 of the Calspan Three-Dimensional Crash Victim Simulation (CVS) Program. Phases I (Bartz, 1971) and II (Bartz et al., 1972) of the CVS program were developed by Calspan under the joint sponsorship of the National Highway Traffic Safety Administration (Contract No. FH-11-7592) and the Motor Vehicle Manufacturers Association (Contract No. 7001-C7). Phase III of the program (Fleck et al., 1974) was sponsored by NHTSA (Contract No. HS-053-2-495) and has been designated as CVS-III, version 11. (Note: The various versions of the CVS program, although numerous, actually represent logical breakpoints in the development of the program made necessary by the NHTSA requirement for frequent distribution of current versions of the program.)

Version 12 of the program was developed (Fleck et al., 1975) under the sponsorship of the Aerospace Medical Research Laboratory (Contract No. F33615-75-C-5002). This version became the basis of the original Articulated Total Body (ATB) model and several of its features (harness-belt systems, wind forces and new joint formulations) were not incorporated into the succeeding versions of the CVS program.

Version 13 of the CVS program incorporated many minor revisions, modifications, additions and corrections (identified at Calspan or reported by users of the program) that did not affect the numerical results of most applications of the program.

Version 14 represented an attempted new belt algorithm under study at Calspan. Although not successful, some of its features were helpful in the development of the new harness-belt algorithm resulting from the current research effort.

Versions 15-18 of the CVS program were developed under the sponsorship of NHTSA (Contract No. HS-6-01300). Version 15 featured the new variable step Vector Exponential Integrator that greatly improved the accuracy while decreasing the required computer run time for the program. This new integrator was supplied to AMRL and was incorporated into the ATB Model.

Version 16 incorporated the new initial positioning equilibrium routine. In version 17, integration of the vehicle and air bag motions was added to the program integrator, more efficient matrix multiplication was developed, improvements were made to the initial direction cosine matrix computation and to the contact routines, and a new output tape for post-processing routines was developed.

Version 18 eliminated the necessity of the multiple output units for the printed tabular time histories that taxed the capacity of many computer systems. The tabular time histories can now be generated (in the same or subsequent run) from the new output unit No. 8 that can also be used to generate user-specified plots of data contained in the tabular time histories. Also, an improved Euler joint algorithm and improved methods of specifying the initial angular orientation of the body segments were developed.

One of the underlying objectives of the present research effort is to combine the features of both the ATB and CVS models into a single program entity. This involved first combining those routines of version 12, the original ATB model, into the current CVS program and then incorporating the new analytical algorithms of this research effort into a new version 19 of the CVS program. This program, as delivered to AMRL, will be designated as the ATB-II model.

It should be emphasized at this point that the above changes to the program, which included many new input options, were incorporated into the CVS program in such a manner that all previous input decks are still acceptable as proper input to the program. In most cases, the original input options are still available as default options so as not to invalidate former

input decks. This is also true for the new input options incorporated into version 19 except that some of the routines of version 12, as they were re-incorporated into the current CVS program, have resulted in some input format changes for the ATB-II model. These changes are well documented in the new input description for the program.

The concept of a harness was introduced in an earlier version of the program to produce the first ATB model (Fleck, 1975). A harness consists of from one to several belts (Figure 1). Each belt is defined as a set of straight line segments connecting reference points. The reference points, P_k, are selected from a prescribed set of points by an algorithm that ensures that the net force of the belt on the segment is directed along the inward normal to the surface.

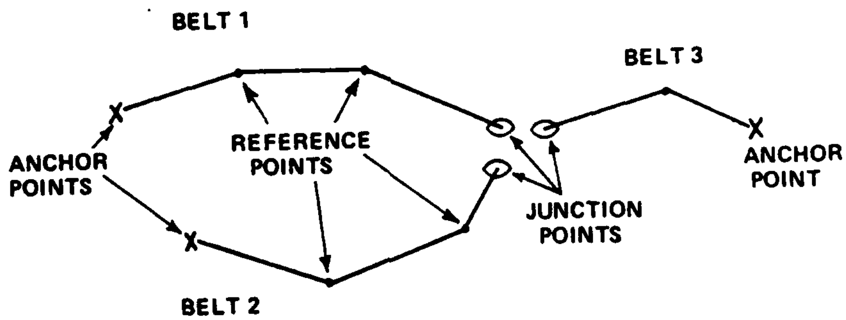


Figure 1 BELT HARNESS MODEL

This early version had three deficiencies. First, interactive belts were tied to a common segment; this segment was moved dynamically to balance the forces. Tests showed that the integrator required small step sizes to prevent oscillation. Second, uniform tension in each belt was assumed. No allowance was made for the effects of friction. Third, reference points were fixed with respect to the segment. The belt could neither penetrate (deform) nor slide along the surface.

The advanced harness-belt system developed for the ATB-II model corrects for all three of these deficiencies. Belts are coupled by the use of Lagrange multipliers; finite friction is introduced; and deformation of the segment is allowed.

DESCRIPTION OF THE MODEL

A belt is represented as a series of straight line segments connecting points P_k, $k = 1, N$. The geometry of the local coordinate reference system for each point is depicted in Figure 2.

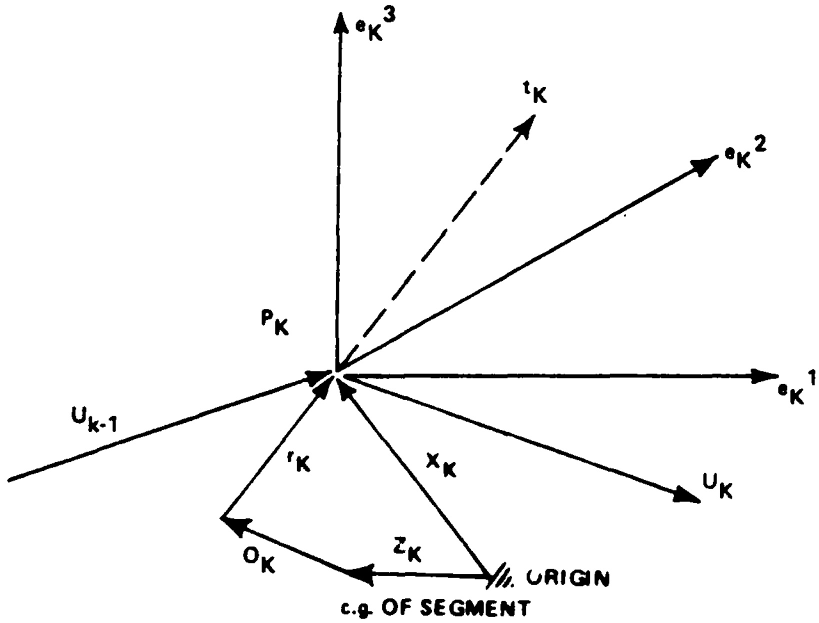


Figure 2 GEOMETRY OF A POINT FOR ADVANCED HARNESS-BELT SYSTEM

The location of each point is given by the vector X_k. Each point is considered to be attached to some segment and is located in the segment reference system by an offset vector O_k, which may be zero, and a reference vector r_k which must not be equal to zero. The offset vector is defined in the local reference system of the segment and is held constant. The reference vector r_k is also defined in the local coordinate system of the segment and is perturbed (changed) if necessary to satisfy the constraint relations.

The location X_k of the point P_k in inertial reference is given by:

$$X_k = Z_k + D_k^{-1} (O_k + r_k)$$

where Z_k is the location of the cg of the segment in inertial reference and D_k is the direction cosine matrix which defines the orientation of the segment. The vectors U_k are unit vectors defining the belt line. They are computed as

$$U_k = (X_{k+1} - X_k)/b_k \text{ for } k = 1, N-1$$

where $b_k = |X_{k+1} - X_k|$.

Each point may have an associated ellipsoid defined by the matrix E_k and may have a preferred plane vector t_k. If E_k and/or t_k are given, they are defined in the local coordinate system of the segment and are assumed constant.

An orthonormal coordinate system defined by the matrix e_k, with component vectors (e_k^1, e_k^2, e_k^3) is established at each point in the following manner. The outward normal to the surface, e_k^3, is computed as

$$e_k^3 = D_k^{-1} E_k r_k, \text{ if an ellipsoid is given, or}$$
$$e_k^3 = D_k^{-1} r_k \text{ if no ellipsoid is given .}$$

If the surface is deformable, r_k may be perturbed along the normal e_k^3. The second component e_k^1 is computed as

$$e_k^1 = (U_k + U_{k-1}) \otimes e_k^3 \text{ , if no preferred plane is given,}$$
$$e_k^1 = (D_k^{-1} t_k) \otimes e_k^3 \text{ , if a preferred plane vector } t_k \text{ is given.}$$

Note that e_k^1 is perpendicular to the outward normal e_k^3 in both cases.

In the first case, $U_k + U_{k-1}$ is the average belt line, hence e_k^1 is perpendicular to the average belt line. At the end points $e_1^1 = U_1 \otimes e_1^3$ and $e_N^1 = U_{N-1} \otimes e_N^3$ since neither U_0 or U_N is defined. There is a possibility that for the end points these may be zero. To insure against this possibility, it is recommended that a preferred plane vector t_k to be defined for the end points where t_k is not parallel to r_k.

In cases of finite friction r_k may be perturbed along e_k^2. In all cases the third component e_k^2 is defined to produce a right handed coordinate system as:

$$e_k^2 = e_k^3 \otimes e_k^1$$

The three component vectors are then normalized by dividing by their magnitude $|e_k^j| = (e_k^j \cdot e_k^j)^{1/2}$. When the belt slips in the direction of the vector e_k^2, it is impossible to distinguish between a redefinition of the basic reference length, B_k, of the belt between points (distance between material points on the belt) and a perturbation of the reference point in this direction (the reference point tracks a material point on the belt). Hence an option is given (called type 5). If a point is type 5, the constraint equations are solved allowing r_k to be perturbed along e_k^2. If a point is not type 5, the constraint equations are solved by perturbing the basic reference length, B_k, of the belt along e_k^2.

Constraint Equations

The belt lies along the vectors $X_{k+1} - X_k$.

The strain in the belt is $b_k/B_k - 1$ where $b_k = |X_{k+1} - X_k|$ and B_k is the current reference length of the kth section of the belt. The B_k's are initially set (at time = 0) by computing the b_k and then adjusting for a user specified initial slack or strain uniformly along the belt.

The stress in the belt is

$$fb_k = fb_k\,(b_k/B_k - 1)$$

where fb_k is a user specified stress-strain function. The current version of the program uses the same functions for all the sections of a particular belt. The function may be modified for strain rate effects where the strain rate is defined as the instantaneous value of $\dot{b}_k/B_k$ (see the section on rate dependent functions).

The belt force at point P_k is

$$FB_k = fb_k\,U_k - fb_{k-1}\,U_{k-1}$$

This force is resolved into its components in the reference system e_k as

$$FR_j = e_k^j \cdot FB_k \text{ for } j = 1, 2, 3.$$

(FR_j is dependent on k but for simplicity of notation the k subscript is not used.)

The perturbation δp_k at the point P_k has three components:

- a perturbation of r_k along e_k^1 of $e_k^1 \cdot \delta p_k$ ($\perp$ belt line, $\perp$ normal)
- a perturbation of B_k along e_k^2 of $e_k^2 \cdot \delta p_k$ (along belt line)
- a perturbation of r_k along e_k^3 of $e_k^3 \cdot \delta p_k$ (normal to surface)

The total perturbation of r_k is then

$$\delta r_k = e_k^1\, e_k^1 \cdot \delta p_k + e_k^3\, e_k^3 \cdot \delta p_k$$

The perturbation of B_k due to slippage at P_k is

$$\delta B_k = e_k^2 \cdot \delta p_k = \delta \ell_k$$

The total change in B_k is

$$\delta B_k = \delta \ell_k - \delta \ell_{k+1}$$

If a point is defined to be type 5 (no belt slippage), then $\delta \ell_k = 0$ and r_k is perturbed along e_k^2 thus,

$$\delta r_k = e_k^1 \, e_k^1 \cdot \delta p_k + e_k^2 \, e_k^2 \cdot \delta p_k + e_k^3 \, e_k^3 \cdot \delta p_k = \delta p_k$$

The three constraint equations which must be satisfied are:

$$|FR_1| < \mu_1 \, |FR_3|$$

$$|FR_2| < \mu_2 \, |FR_3|$$

$$fD_k \, (\rho_k) = |FR_3|$$

where μ_1, μ_2 are coefficients of friction (constant) and $fD_k(\rho_k)$ is the force deflection function for the belt segment interaction at the point P_k. The deflection parameter ρ_k is defined later. This function may be modified by the penetration rate $\dot{\rho}_k$ (see section on rate dependent functions).

The following special cases are allowed.

If no friction function is defined, the friction coefficients are assumed to be infinite and the first two constraints are automatically satisfied. No perturbation will occur along e_k^1 or e_k^2.

If no force deflection function is specified, the segment is considered to be undeformable and the third constraint is automatically satisfied. No perturbation will occur along e_k^3.

The penetration parameter ρ_k is defined as:

$$\rho_k = [\frac{1}{(r_k \cdot E_k r_k)^{1/2}} - 1] |r_k|$$

If the point r_k is outside of the ellipsoid, ρ_k is negative. If the point is on the ellipsoid , ρ_k is zero and if the point is inside of the ellipsoid, ρ_k is positive. If the ellipsoid is a sphere, ρ_k is a direct measure of the penetration.

No provision has been made in the program for the case of $r_k = 0$ or the case where the penetration exceeds the half way point. It is assumed that the force deflection function will be defined to prevent the occurrence of these cases.

SOLUTION OF THE CONSTRAINT EQUATIONS

Each harness (collection of one or more belts) is treated as a unit to allow interaction of the belts. The points, X_k, will be considered sequentially to generate the matrix. A Newton Raphson technique will be used, where the equations are linearized to form a linear set of simultaneous equations in the perturbations δp_k. Interaction between belts is achieved by using Lagrange multipliers to constrain common points (junction points) to be identical. The typical matrix representing the simultaneous equations will be of the form shown in Figure 3.

$$
\begin{bmatrix}
 & & I & & & -I & & \\
 & X & X & & & & & \\
I & X & X & X & & & & \\
 & & X & X & X & & & \\
 & & & X & X & & & \\
-I & & & & & X & X & \\
 & & & & & X & X & X \\
 & & & & & & X & X
\end{bmatrix}
\begin{bmatrix}
\delta\lambda_1 \\ \delta p_1 \\ \delta p_2 \\ \delta p_3 \\ \delta p_4 \\ \delta p_5 \\ \delta p_6 \\ \delta p_7
\end{bmatrix}
=
\begin{bmatrix}
0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7
\end{bmatrix}
$$

I 3 x 3 IDENTITY MATRIX

X NON ZERO 3 x 3 ENTRY

$\delta\lambda_1$ LAGRANGE MULTIPLIER

δp_k PERTURBATIONS

y_k RIGHT HAND SIDE (ZERO WHEN CONVERGED)

Figure 3 MATRIX FORM OF CONSTRAINT EQUATIONS

A sample belt system, illustrated in Figure 4, has points 2 and 5 which are common. Since the equations produce a sparse matrix, the technique used to solve the system equations in the CVS program will be used where the 3 x 3 sub-matrices are generated and subroutine FSMSOL is called to solve the system.

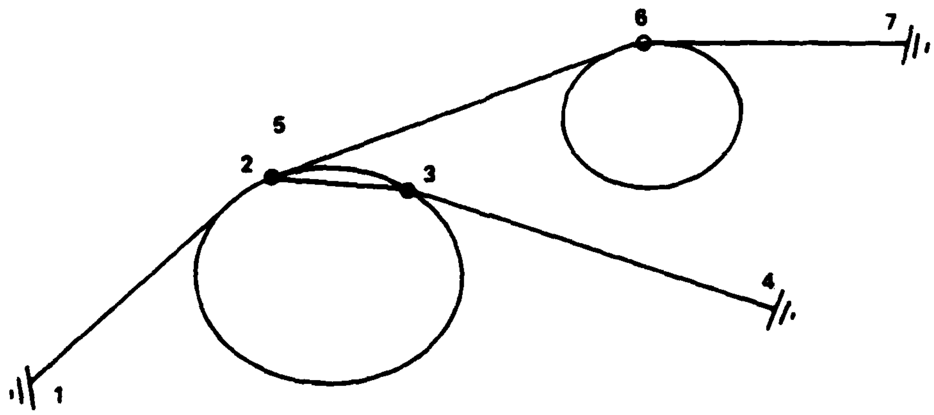


Figure 4 POINTS OF A SAMPLE BELT SYSTEM

PERTURBATION EQUATIONS

In the subsequent development, the 3 x 3 sub-matrices are denoted by C_{jk} where j identifies the row and k the column.

The three constraint equations for the kth joint may be written as:

$$|e_k^j \cdot FB_k| - \mu_j \, |e_k^3 \cdot FB_k| \leq 0 \qquad \text{for } j = 1, 2$$

$$fD_k \, (\rho_k) - |e_k^3 \cdot FB_k| = 0$$

Each of these is a scalar equation. If the constraints are not satisfied, we write the perturbation equations as a vector:

$$e_k^j \left\{ \delta[\mu_j \, |e_k^3 \cdot FB_k| - |e_k^j \cdot FB_k|] \right\} = e_k^j \left\{ |e_k^j \cdot FB_k| - \mu_j \, |e_k^3 \cdot FB_k| \right\},$$

$$\text{for } j = 1, 2$$

$$e_k^3 \left\{ \delta[|e_k^3 \cdot FB_k| - fD_k \, (\rho_k)] \right\} = e_k^3 \left\{ fD_k \, (\rho_k) - |e_k^3 \cdot FB_k| \right\}$$

It will be shown that these are of the form

$$e_k^j \, V_k^j \cdot \delta\rho_k = e_k^j \, r_k^j \qquad \text{for } j = 1, 2, 3$$

where V_k^j is a vector, r_k^j is a scalar and $\delta\rho_k$ are the perturbations (the perturbation equations are linearized). The three equations are summed to form:

$$[e_k^1 \, V_k^1 \cdot + e_k^2 \, V_k^2 \cdot + e_k^3 \, V_k^3 \cdot] \, \delta\rho_k = e_k^1 \, r_k^1 + e_k^2 \, r_k^2 + e_k^3 \, r_k^3$$

This procedure thus forms a matrix equation. In general, the perturbation of adjacent points affect the constraint equations at the kth point because of the linkage from the belt. Thus, when considering the kth point, we are generating entries in the kth row of the set of simultaneous equations.

This row represents the equation:

$$C_{k,k-1}\ \delta\rho_{k-1} + C_{k,k}\ \delta\rho_k + C_{k,k+1}\ \delta\rho_{k+1} = C_k$$

where $C_{k\ell} = e_k^1\, V_\ell^1 \cdot + e_k^2\, V_1^2 \cdot + e_k^3\, V_\ell^3 \cdot$ for $\ell = k-1,\ k,\ k+1$

$$C_k = e_k^1\, r_k^1 + e_k^2\, r_k^2 + e_k^3\, r_k^3$$

where $C_{k\ell}$ is a 3 x 3 matrix and C_k is a 3 vector (for interactive belts this row may be displaced by the number of Lagrange multipliers used). This procedure of forming a matrix equation from a set of scalar equations is valid since the e_k^j form an orthogonal set.

If a constraint is satisfied, we set

$$V_k^j = e_k^j, \quad r_k^j = 0, \quad \text{and} \quad V_{k-1}^j = V_{k+1}^j = 0$$

If all three constraints are satisfied, this will yield

$$I\ \delta\rho_k = 0$$

since $I = e_k^1\, e_k^1 \cdot + e_k^2\, e_k^2 \cdot + e_k^3\, e_k^3 \cdot$

NOTE: If no friction function is specified, infinite friction is assumed and constraints one and two are assumed to be satisfied. If no force deflection function is specified, the segment is assumed to be nondeformable and constraint three is assumed to be satisfied.

DETAILS OF PERTURBATION EQUATIONS

Consider the perturbation equation at point k

$$\delta\ [\mu_j\ |e_k^3 \cdot FB_k| - |e_k^j \cdot FB_k|] = |e_k^j \cdot FB_k| - \mu_j\ |e_k^3 \cdot FB_k| \quad \text{for } j = 1,2$$

we have

$$\delta|e_k^i \cdot FB_k| = (\delta e_k^i \cdot FB_k + e_k^i \cdot \delta FB_k) \text{ sign } (e_k^i \cdot FB_k) \quad \text{for } i = 1,2,3$$

$$FB_k = fb_k U_k - fb_{k-1} U_{k-1}$$

$$U_k = (X_{k+1} - X_k)/b_k$$

$$b_k = |X_{k+1} - X_k|$$

$$fb_k = fb_k (b_k/B_k - 1) \quad \text{(stress - strain)}$$

$$\delta U_k = (I - U_k U_k \cdot) (\delta X_{k+1} - \delta X_k)/b_k$$

$$\delta fb_k = fb_k' (\delta b_k/B_k - b_k \delta B_k/B_k^2), \quad fb_k' = \frac{\delta}{\delta S} fb_k(S)$$

$$\delta B_k = \delta \ell_k - \delta \ell_{k+1}$$

$$\delta b_k = U_k \cdot [\delta X_{k+1} - \delta X_k]$$

$$\delta X_k = D_k^{-1} \delta r_k$$

$$\delta e_k^3 = (I - e_k^3 e_k^3 \cdot) D_k^{-1} E_k \delta r_k / |E_k r_k| , \quad \text{if ellipsoid given}$$

$$= (I - e_k^3 e_k^3 \cdot) D_k^{-1} \delta r_k / |r_k| , \quad \text{if no ellipsoid}$$

(this is the same as if $E_k = I$, a unit sphere)

$$\delta e_k^1 = (I - e_k^1 e_k^1 \cdot) [(\delta U_k + \delta U_{k-1}) \otimes e_k^3 + (U_k + U_{k-1}) \otimes \delta e_k^3] / |(U_k + U_{k-1}) \otimes e_k^3|$$

or if a preferred direction t_k is specified

$$\delta e_k^1 = (I - e_k^1 e_k^1 \cdot) (D_k^{-1} t_k) \otimes \delta e_k^3 / |(D_k^{-1} t_k) \otimes e_k^3|$$

in all cases

$$\delta e_k^2 = (\delta e_k^3) \otimes e_k^1 + e_k^3 \otimes \delta e_k^1$$

Collecting the terms for δFB_k we have:

$$\delta FB_k = [I\ fb_k/b_k + U_k\ U_k\cdot\ (fb'_k/B_k - fb_k/b_k)]\ (D^{-1}_{k+1}\ \delta r_{k+1} - D^{-1}_k\ \delta r_k)$$

$$- U_k\ (fb'_k\ b_k/B^2_k)\ (\delta\ell_k - \delta\ell_{k+1})$$

$$- [I\ fb_{k-1}/b_{k-1} + U_{k-1}\ U_{k-1}\cdot\ (fb'_{k-1}/B_{k-1} - fb_{k-1}/b_{k-1})]$$

$$(D^{-1}_k\ \delta r_k - D_{k-1}\ \delta r_{k-1})$$

$$+ U_{k-1}\ (fb'_{k-1}\ b_{k-1}/B^2_{k-1})\ (\delta\ell_{k-1} - \delta\ell_k)$$

Defining the matrices A_k, A_{k-1}

$$A_k = (fb_k/b_k)\ e_k\cdot + e_k \cdot U_k\ U_k \cdot (fb'_k/B_k - fb_k/b_k)$$

$$A_{k-1} = (fb_{k-1}/b_{k-1})\ e_k\cdot + e_k \cdot U_{k-1}\ U_{k-1} \cdot (fb'_{k-1}/B_{k-1} - fb_{k-1}/b_{k-1})$$

and the vectors S_k, S_{k-1} as

$$S_k = e_k \cdot U_k\ fb'_k\ b_k/B^2_k$$

$$S_{k-1} = e_k \cdot U_{k-1}\ fb'_{k-1}\ b_{k-1}/B^2_{k-1}$$

we may write the $e^j_k \cdot FB_k$ for j = 1,2,3 as the components of the vector

$$e_k \cdot \delta FB_k = A_k\ D^{-1}_{k+1}\ \delta r_{k+1} + S_k\ \delta\ell_{k+1}$$

$$- (A_k + A_{k-1})\ D^{-1}_k\ \delta r_k - (S_k + S_{k-1})\ \delta\ell_k$$

$$+ A_{k-1}\ D^{-1}_{k-1}\ \delta r_{k-1} + S_{k-1}\ \delta\ell_{k-1}$$

(e_k is a 3 x 3 matrix, $e_k\cdot$ is the transpose of e_k) the dot notation is used for transposes of vectors and matrices to eliminate confusion with superscripts.

In the case of a 'type 5' point $D^{-1}_k\ \delta r_k = \delta\rho_k$

For a general point

$$D_k^{-1}\ \delta r_k = (e_k^1\ e_k^1 \cdot + e_k^3\ e_k^3 \cdot)\ \delta\rho_k$$

$$\delta \ell_k = e_k^2 \cdot \delta\rho_k$$

The terms for the jth constraint may be collected as

$$\mu_j\ e_k^3 \cdot FB_k \text{ sign } (e_k^3 \cdot FB_k) - e_k^j \cdot FB_k \text{ sign } (e_k^j \cdot FB_k)$$

$$= V_{k+1}^j \cdot \delta\rho_{k+1} + V_k^j \cdot \delta\rho_k + V_{k-1}^j \cdot \delta\rho_{k-1}$$

In this version we collect only the terms due to the δFB_k and ignore the perturbations of the coordinate system δe_k which we believe will be small compared to the variation of the belt forces. Since the e_k are recomputed for each iteration, this is a valid procedure providing it does not cause problems with convergence of the iterations. It is somewhat analogous to the solution of the equation $x = f(x)$ by the iterative sequence $x_{n+1} = f(x_n)$, $x_n \rightarrow x$ if convergence is obtained.

The third constraint is handled in a similar fashion. It requires the evaluation of $\delta fD_k\ (\rho_k)$. Since ρ_k has been defined as

$$\rho_k = (1/\sqrt{r_k \cdot E_k\ r_k} - 1)\ |r_k|$$

we have

$$\delta fD_k\ (\rho_k) = fD_k'\ \delta\rho_k$$

where $fD_k' = \dfrac{\delta fD_k}{\delta\rho_k}$, evaluated at ρ_k

$$\delta\rho_k = (\rho_k\ r_k \cdot /(r_k \cdot r_k) - |r_k|\ (E_k\ r_k) \cdot /(r_k \cdot E_k\ r_k)^{3/2})\ \delta r_k$$

Thus, for the third constraint, we collect the terms of

$$\delta(|e_k^3 \cdot FB_k| - fD_k\ (\rho_k)) = fD_k\ (\rho_k) - |e_k^3 \cdot FB_k|$$

RATE DEPENDENT FUNCTIONS AND ENERGY LOSS

Forces in the harness-belt system can be produced by the stress-strain in the belt segments and by penetration of the body segments by the belt. Both of these forces are computed within the ATB-II model by the force deflection functions that are specified as input. Previously, these functions were basically static with provisions for initial loading, unloading and reloading through use of inertial spike, energy absorption and permanent deflection functions.

In order to include dynamic effects, the total force deflection function may be computed assuming the functional form

$$f(\rho, \dot{\rho}) = f_1 (\rho) + f_2 (\dot{\rho}) + f_3 (\dot{\rho}) \times f_4 (\rho)$$

where ρ is the penetration or strain, $\dot{\rho}$ is the time derivative of ρ and f_j for j = 1, 4 are the standard type functions which may be defined as polynomial, tabular or constant functions. f_2, f_3 and f_4, if used, replace the normal use of the initial loading, energy absorption and permanent deflection functions to compute the total force deflection function within the program. The use of these rate dependent functions to specify a decreased unloading function can be accomplished by taking into account that $\dot{\rho}$ is negative during unloading.

When work is done during the loading, unloading cycles the part that is attributed to $f_1 (\rho)$ may be considered recoverable (potential) energy. The remainder is lost. Thus, the rate of energy loss is given by

$$\dot{e}_{loss} = (f_2 (\dot{\rho}) + f_3 (\dot{\rho}) f_4 (\rho)) \times \dot{\rho}$$

(note that for loss, $\dot{\rho} f_2 (\dot{\rho})$ and $\dot{\rho} f_3 (\dot{\rho}) \times f_4 (\rho)$ shoud be non negative.)

The energy loss is thus

$$e_{loss} = \int_0^{\tau} \dot{e}_{loss}\, dt$$

The current version of the program has made no provision for the integration of auxiliary variables hence a simple integration is performed by storing $\dot{e}_{loss}$ in each contact routine and at update time (completion of an integration step) computing $\Delta e_{loss} = \dot{e}_{loss} \times h$ where h is the step size of the just completed successful integration step. The Δe_{loss} are accumulated as an approximation for e_{loss}.

RATE CALCULATION FOR HARNESS ROUTINES

<u>Strain Rate</u>

The strain in the kth belt section is given by

$$S = b_k/B_k - 1$$

where $b_k = |X_{k+1} - X_k|$ and B_k is the unstrained length of the belt in this section. Taking the time derivatives, we have

$$\dot{S} = \dot{b}_k/B_k - (b_k/B_k^2)\, \dot{B}_k$$

Now

$$b_k = |X_{k+1} - X_k|$$

$$\dot{b}_k = (X_{k+1} - X_k) \cdot (\dot{X}_{k+1} - \dot{X}_k)/b_k = U_k \cdot (\dot{X}_{k+1} - \dot{X}_k)$$

$$X_j = Z_j + D_j^{-1} (r_j + O_j)$$

$$\dot{X}_j = \dot{Z}_j + D_j^{-1} \omega_j \otimes (r_j + O_j) + D_j^{-1} \dot{r}_j , \quad j = k, k+1$$

where $\dot{Z}_j$ is the linear velocity at the cg and ω_j is the angular velocity of the segments associated with point k (D_j is the direction cosine matrix). During the integration step r_k and B_k are held fixed. The values of $\dot{r}_k$ and $\dot{B}_k$ are estimated at the end of the perturbation routine by

$$\dot{r}_k \simeq \frac{\delta r_k}{h} = \frac{\text{new } r_k - \text{previous } r_k}{h}$$

$$\dot{B}_k \simeq \frac{\delta B_k}{h} = \frac{\text{new } B_k - \text{previous } B_k}{h}$$

where h is the integration step size used in the last successful integration step.

Penetration Rate

Penetration ρ_k is defined as

$$\rho_k = (1/\sqrt{r_k \cdot E_k r_k} - 1)|r_k|$$

where r_k is the reference vector and E_k is the ellipsoid matrix associated with point k. Taking the time derivative we have

$$\dot{\rho}_k = \frac{\rho_k}{r_k \cdot r_k} r_k \cdot \dot{r}_k - \frac{|r_k| (E_k r_k) \cdot \dot{r}_k}{(r_k \cdot E_k r_k)^{3/2}}$$

where

The value of $\dot{r}_k$ is estimated at the end of the perturbation routine as $\dot{r}_k \simeq \delta r_k/h$ and is held fixed during the next integration step. Thus, $\dot{\rho}_k$ will not change during the course of an integration step (unless the program logic is changed to call the perturbation routine during the course of an integration step.)

OMNI-DIRECTIONAL SPECIFIED MOTION

The program had been modified to allow the specification of the motion of up to six segments. In previous versions of the program only the motion of the vehicle could be specified. The segments are arbitrary. If the motion of two or more segments, which are in the same tree structure, are specified the user must be sure that these motions are compatible with the structure.

Since the belt tiedown points may be attached to any segment, this satisfies the requirement that the motion of the tiedown points may be specified.

The previous options of a one-half sine wave unidirectional deceleration pulse, a unidirectional tabular deceleration table or the omni-directional (6 degree of freedom) are available.

SPLINE FIT METHOD

The omni-directional input has been modified to provide (as an option) the capability of inputting position, velocity or acceleration data at specified time points. The time points need not be equally spaced. The program will produce an equally spaced acceleration table. The following procedure is used.

From the given data for each of the six components (three linear, three angular) a polynomial (0 to 3rd degree) fit is computed. The polynomial is then used to fill out an equally spaced acceleration table for that component.

The degree of the polynomial is optional with the user; however, if position data is supplied, only a quadratic or cubic fit should be used since the second derivative must be defined. If velocity data is specified a linear, quadratic, or cubic fit may be used and if acceleration data is specified, the constant, linear, quadratic or cubic may be used.

The constant fit is continuous on the right, i.e., if three points are specified (t_1, X_1), (t_2, X_2), (t_3, X_3) the functional fit is:

$$f_1(t) = X_1,\ t < t_2$$

$$f_2(t) = X_2,\ t_2 \leq t < t_3$$

$$f_3(t) = X_3,\ t_3 \leq t$$

If a linear fit is specified, the fit is:

$$f_1(t) = X_1 + \frac{X_2 - X_1}{t_2 - t_1}(t - t_1),\ t < t_2$$

$$f_2(t) = X_2 + \frac{X_3 - X_2}{t_3 - t_2}(t - t_2),\ t_2 \leq t < t_3$$

$$f_3(t) = X_3 \qquad ,\ t_3 \leq t$$

Note the function is continuous at the interior time joints (in this case there is only one interior time point at t_2).

If a quadratic is specified the fit is:

$$f_1(t) = X_1 + b_1 (t - t_1) + c_1 (t - t_1)^2,\ t < t_2$$

$$f_2(t) = X_2 + b_2 (t - t_2) + c_2 (t - t_2)^2,\ t_2 \leq t < t_3$$

$$f_3(t) = X_3 \qquad ,\ t_3 \leq t$$

The b's and c's are chosen so that the function and its first derivative are continuous at the interior time points (only one in the above example) and the sums of the squares of the changes in the 2nd derivative is minimized (in this case $(c_1-c_2)^2$). This minimization has the feature that if the points lie on a quadratic, the quadratic will be reproduced.

If a cubic fit is specified, the fit is:

$$f_1(t) = X_1 + b_1\ (t-t_1) + c_1\ (t-t_1)^2 + d_1\ (t-t_1)^3,\ t < t_2$$

$$f_2(t) = X_2 + b_1\ (t-t_2) + c_2\ (t-t_2)^2 + d_2\ (t-t_2)^3,\ t_2 \leq t < t_3$$

$$f_3(t) = X_3 \qquad\qquad ,\ t_3 \leq t$$

The b's, c's and d's are chosen so that the function is first and 2nd derivatives are continuous at the interior joints and the sums of the square of the changes in the 3rd derivatives are minimized.

This minimization will reproduce a cubic. Note for the quadratic and cubic, at least three points must be given. For values of t greater than the last time point, the function is treated as a constant equal to the last value (i.e., X_3 in the example). For values of t less than the first time point the first function is extrapolated.

It is assumed the user will specify enough values to span the range of interest so that extrapolation is not a problem. The first time point must be at time equal to zero.

MODEL INPUT OPTIONS

If position data is given, a quadratic or cubic fit must be specified. The initial position of the cg of the segment is set to the linear data. The angular position is set to the angular data, integrating the data as the yaw, pitch and roll angles.

The initial velocities are set to the values determined from derivative of the spline fit evaluated at t = 0. For the angular data roll rate is interpreted as the angular velocity on the segment local X axis, pitch rate as local Y axis and yaw rate as local Z axis.

The acceleration table is computed from the 2nd derivatives of the spline fit at equally spaced time points.

If velocity data is given, the user also specifies the initial position and orientation. The initial velocities as set to the first point in the velocity table and the acceleration table computer from the polynomial fit (at least linear).

If acceleration data is given, the user also specifies the initial positions and velocities and the acceleration table is computed from the polynomial fit.

A word of caution. The angular information is assumed to be in the segment reference system, except for the yaw, pitch, and roll at time equal to zero which is interpreted as measured in the inertial reference. This should cause no problems except in the case where the tables are generated from positional data and more than one set of non-zero angular information is given. In this case, extreme care must be taken so that the proper interpretation is made.

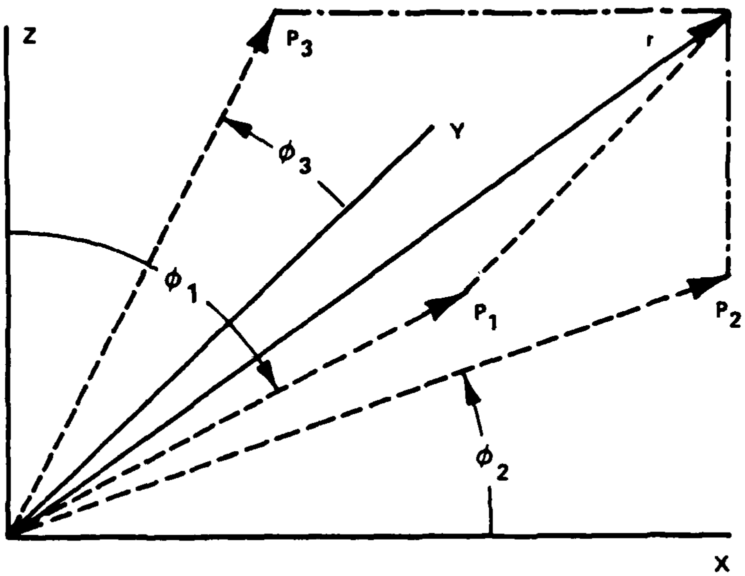


Figure 5 PROJECTION ANGLES OF A SEGMENT AXIS

Consider a vector r and its projections p_1, p_2 and p_3 on the coordinate frame ZX, XY and YZ, respectively. Let ϕ_j be the angles the projections make with the axis as defined in Figure 5. (This is done in accordance with the right hand rule.)

The direction of the vector r may be determined from any two of the three angles. The following table illustrates the proportionality of the components r_x, r_y and r_z of the vector r and the pairs of angles.

Table 1

Projections/	r_x	r_y	r_z
ZX-XY	$\lvert\sin\phi_1\rvert \cos\phi_2$	$\lvert\sin\phi_1\rvert \sin\phi_2$	$\cos\phi_1 \lvert\cos\phi_2\rvert$
XY-YZ	$\cos\phi_2 \lvert\cos\phi_3\rvert$	$\lvert\sin\phi_2\rvert \cos\phi_3$	$\lvert\sin\phi_2\rvert \sin\phi_3$
YZ-ZX	$\lvert\sin\phi_3\rvert \sin\phi_1$	$\cos\phi_3 \lvert\cos\phi_1\rvert$	$\lvert\sin\phi_3\rvert \cos\phi_1$

A unit vector in the direction of r may be established by computing the components as given in the table and then normalizing the vector.

Determination of Direction Cosine Matrix

The orientation of a rigid body is specified by its direction cosine matrix. This matrix can be computed from the orientations of two of the three principle axes of the body because the principle axes are orthogonal and given two, the third is prescribed to form a right-handed coordinate system.

The program has been modified to allow the user to specify the orientation of any segment by the following procedure.

1. Define the orientation of a primary principle axes (X, Y or Z of the rigid body) by the projection method or by the component method.

   In the projection method the user inputs two angles and identifies the projection pair that these angles refer to (i.e., common X; ZX-XY projections, common Y or common Z).

   In the component method the user inputs the three components (r_x, r_y, r_z) which specify the orientation. These need not form a unit vector.

The principle axis which this primary vector is describing is identified.

2. In the same fashion a secondary principle axis is defined. The program will then compute the direction cosine matrix.

The reason for defining a primary and a secondary axis is the inability to input precise data. To overcome this the program assumes the primary axis is precise. That is, it normalizes the vector without other modification. It then modifies the secondary axis to make it perpendicular to the primary and a unit vector. The remaining principle axis is computed to form a right-handed system.

APPENDIX A

THE INPUT DESCRIPTION FOR
THE AMRL ARTICULATED TOTAL BODY (ATB-II) MODEL

The input description describes those features that are operational through version 19 of the CVS program. The ATB-II model was developed for the Air Force Aerospace Medical Research Laboratory, Wright-Patterson Air Force Base, Ohio 45433 under Contract No. F33615-78-C-0516. It contains what Calspan considers to be the best description of the program capabilities.

```
INPUT DESCRIPTION FOR THE AMRL ARTICULATED TOTAL BODY (ATB-II) MODEL     |
                         30 OCTOBER 1979                                 |

NOTE: THIS REPORT IS SUPPLIED WITH '1' IN COLUMN 1 FOR PAGE SKIP         |
      CONTROL TO ALLOW FOR PRINTING ON VARIOUS COMPUTER SYSTEMS.         |

      THE FOLLOWING SPECIAL SYMBOLS MAY DIFFER ON OTHER SYSTEMS:         |

           "#" IS USED TO INDICATE "NOT EQUAL".                          |
           "<" IS USED TO INDICATE "LESS THAN".                          |
           ">" IS USED TO INDICATE "GREATER THAN".                       |
           "|" IS USED TO INDICATE "ABSOLUTE VALUE".                     |

ANY LINE WITH EITHER OF THE SYMBOLS "|","*" OR "$" AT THE RIGHT          |
INDICATES THAT A CHANGE HAS BEEN MADE TO THIS INPUT DESCRIPTION          |
SINCE THAT INCLUDED IN NHTSA REPORT NOS. DOT-HS-801 507 THROUGH 510,     |
"AN IMPROVED THREE DIMENSIONAL COMPUTER SIMULATION OF MOTOR VEHICLE      |
CRASH VICTIMS", APRIL 1975 (FORMERLY CALSPAN REPORT NO. ZQ-5180-L-1)     |

THE SYMBOL "*" INDICATES THAT AN ITEM OR CARD HAS BEEN ADDED TO THE      |
CVS MODEL INPUT IN SUCH A MANNER THAT PREVIOUS INPUT DECKS ARE STILL     |
ACCEPTABLE AS PROPER INPUT FOR THE CURRENT VERSION OF THE PROGRAM.       |

THE SYMBOL "$" INDICATES THAT CHANGES IN FORMAT OR CONTENT ARE           |
REQUIRED TO PREVIOUS INPUT DECKS TO BE ACCEPTABLE AS PROPER INPUT        |
FOR THE CURRENT VERSION OF THE PROGRAM.                                  |


OUTLINE OF INPUT TO THE PROGRAM :


CARDS A -  DATE AND RUN DESCRIPTION, UNITS OF INPUT AND OUTPUT,
           CONTROL OF RESTART, INTEGRATOR AND OPTIONAL OUTPUT.

CARDS B -  PHYSICAL CHARACTERISTICS OF THE SEGMENTS AND JOINTS.

CARDS C -  DESCRIPTION OF THE VEHICLE MOTION.

CARDS D -  CONTACT PLANES, BELTS, AIR BAGS, CONTACT ELLIPSOIDS,
           CONSTRAINTS, AND SYMMETRY OPTIONS.

CARDS E -  FUNCTIONS DEFINING FORCE-DEFLECTIONS, INERITIAL SPIKE,
           ENERGY ABSORPTION FACTOR, AND FRICTION COEFFICIENTS.

CARDS F -  ALLOWED CONTACTS AMONG SEGMENTS, PLANES, BELTS, AIR BAGS
           AND CONTACT ELLIPSOIDS.

CARDS G -  INITIAL ORIENTATIONS AND VELOCITIES OF THE SEGMENTS.

CARDS H -  CONTROL OF OUTPUT OF TIME HISTORY OF SELECTED SEGMENT
           MOTIONS AND JOINT PARAMETERS.

CARDS I -  CONTROL INFORMATION FOR PLOTTER OUTPUT                        *
```

DESCRIPTION OF FORTRAN FORMAT STATEMENTS USED

AT THE BEGINNING OF THE DESCRIPTION OF EACH CARD APPEARS THE FORTRAN FORMAT STATEMENT THAT SPECIFIES THE STRUCTURE OF THE INPUT IMAGE FOR THAT CARD. THE ONLY FORMAT CODES USED BY THE CVS PROGRAM ARE

NFW.D	(F TO DESCRIBE REAL DATA FIELDS)
NIW	(I TO DESCRIBE INTEGER DATA FIELDS)
NAW	(A TO DESCRIBE ALPHANUMERIC DATA FIELDS)
WX	(X TO INDICATE A FIELD TO BE SKIPPED)

WHERE: N, W AND D ARE UNSIGNED INTEGER CONSTANTS

N IS OPTIONAL AND IS A REPEAT COUNT USED TO DENOTE THE NUMBER OF TIMES THE FORMAT CODE IS TO BE USED. IF N IS OMITTED, A VALUE OF ONE IS ASSUMED AND THE CODE IS USED ONLY ONCE.

W SPECIFIES THE FIELD WIDTH (NUMBER OF COLUMNS ON THE CARD).

D NORMALLY SPECIFIES THE NUMBER OF DECIMAL PLACES TO THE RIGHT OF THE DECIMAL POINT, I.E., THE FRACTIONAL PART OF THE NUMBER. HOWEVER, A DECIMAL POINT SUPPLIED WITHIN THE FIELD WILL OVER-RIDE THE D SPECIFICATION.

/ IS USED TO INDICATE THE END OF A CARD IMAGE AND THAT THE REMAINING FIELDS ARE TO BE SUPPLIED ON A SUCCEEDING CARD.

ALL VARIABLE NAMES USED FOLLOW THE STANDARD FORTRAN NAMING CONVENTION, I.E., THOSE VARIABLES WHERE THE FIRST LETTER OF THEIR NAME IS A-H OR O-Z ARE REAL (ACTUALLY DOUBLE PRECISION ON IBM AND UNIVAC COMPUTERS AND SINGLE PRECISION ON CDC COMPUTERS) AND THOSE WITH I-N AS THEIR FIRST LETTER ARE INTEGER.

ALL REAL DATA HAVE A FW.0 FORMAT CODE WHICH REQUIRES THE USE OF A DECIMAL POINT WITHIN THE SPECIFIED FIELD TO OVERRIDE THE D=0 SPECIFICATION. ON MOST COMPUTERS F, D AND E FORMAT CODES ARE COMPLETELY INTERCHANGEABLE FOR INPUT WHICH PERMITS ONE TO SUPPLY AN EXPONENTIAL (POWER OF TEN) MULTIPLIER; E.G., 0.000001 MAY BE SUPPLIES AS 1.0D-6, PROVIDED THAT THE EXPONENTIAL TERM IS RIGHT ADJUSTED WITHIN THE FIELD WIDTH. IN ALL OTHER CASES, REAL DATA USING THE FW.0 FORMAT CODE MAY APPEAR ANYWHERE WITHIN THE FIELD WIDTH. ALL BLANKS ARE ASSUMED TO BE A ZERO AND THEREFORE IGNORED. A BLANK FIELD WILL THEREFORE INPUT A VALUE OF ZERO.

ALL INTEGER DATA USE A IW FORMAT CODE AND MUST BE RIGHT ADJUSTED, I.E., MUST APPEAR IN THE RIGHTMOST COLUMNS OF THE FIELD.

SEVERAL NAMES, TITLES AND OTHER DESCRIPTIVE ITEMS ARE ALPHANUMERIC DATA AND USE THE AW FORMAT CODE. HERE BLANKS ARE SPACES AND THE ACTUAL CHARACTERS DESIRED MAY APPEAR ANYWHERE WITHIN THE FIELD.

A. MAIN PROGRAM INPUT

CARD A.1.A	FORMAT (3A4, 2I4, F8.0)
DATE(I),I=1,3	DATE OF THE RUN (12 CHARACTERS).
IRSIN	RESTART INPUT UNIT NO. IF BLANK OR ZERO, ALL INPUT TO BE SUPPLIED ON CARDS A.3 TO CARDS H.7. IF NONZERO (SUGGESTED VALUE =4) INPUT WILL BE SUPPLIED FROM A PREVIOUS RESTART TAPE AND CARDS A.1.B,C AND A.2.
IRSOUT	RESTART OUTPUT UNIT NO. IF NONZERO (SUGGESTED VALUE =3) RECORDS WILL BE WRITTEN ON THIS OUTPUT UNIT FOR FUTURE RESTART RUNS. AN INITIAL RECORD CONTAINING ALL INPUT AND INITIALIZATION DATA WILL BE WRITTEN PLUS A TIME POINT RECORD AT EVERY TIME INTERVAL AS SPECIFIED BY DT ON CARD A.4.
RSTIME	RESTART TIME (SEC.) REQUIRED IF IRSIN ≠ 0. SHOULD BE NONZERO AND AN INTEGER MULTIPLE OF DT ON CARD A.4. PROGRAM WILL READ RECORDS FROM THE PREVIOUS RESTART TAPE UP TO AND INCLUDING THIS TIME, MAKE CHANGES PER CARD A.2 AND CONTINUE OPERATION FROM THERE.
CARDS A.1.B - A.1.C	FORMAT (20A4 / 20A4) **
COMENT(I),I=1,40	DESCRIPTION OF THE RUN (160 CHARACTERS ON TWO CARDS).

** ANY FORMAT MARKED IN THIS MANNER INDICATES THAT COLUMNS 73-80 OF THAT CARD ARE USED FOR INPUT AND SHOULD NOT BE USED FOR IDENTIFICATION.

CARDS A.2 ARE REQUIRED ONLY IF IRSIN > 0, IN WHICH CASE ALL OTHER INPUT AS SPECIFIED ON CARDS A.3 TO H.7 ARE BYPASSED. TWO SETS OF A.2 (EACH TERMINATED WITH A BLANK CARD) ARE REQUIRED. THE FIRST SET IS PROCESSED AFTER THE INITIAL INPUT RECORD IS READ FROM INPUT UNIT IRSIN AND, IF IRSOUT ≠ 0, BEFORE THE INPUT RECORD IS WRITTEN ON OUTPUT UNIT IRSOUT. THE SECOND SET IS PROCESSED AFTER THE TIME POINT RECORD FOR TIME = RSTIME HAS BEEN READ AND, IF IRSOUT ≠ 0, AFTER THE SAME RECORD IS WRITTEN ON OUTPUT UNIT IRSOUT, BUT BEFORE THE PROGRAM RESUMES OPERATION.

CARDS A.2.A - A.2.N	FORMAT(A8, 4I4, 2(F8.0, I8, A8))
AVAR	ALPHANUMERIC NAME (LEFT ADJUSTED IN FIELD) OF VARIABLE TO BE REDEFINED FOR RESTART. PROGRAM IS CAPABLE OF CHANGING ANY VARIABLE IN THE LABELED COMMON BLOCKS AS USED AFTER ALL INITIALIZATION HAS BEEN PERFORMED. THE USER SHOULD ASCERTAIN THAT CHANGING THIS VARIABLE IS VALID FOR THE PROGRAM.
INDEX(I),I=1,3	THE ARRAY INDICES, IF ANY, OF THE VARIABLE. MUST AGREE IN NUMBER AND THE VALUES MUST BE LESS THAN OR EQUAL TO THE DIMENSIONS OF THE VARIABLE. BLANK OR ZERO FOR NO DIMENSION.
ITYPE	SUPPLY 1,2 OR 3 TO INDICATE THAT THE NEW VALUE IS TO BE REAL(RR), INTEGER(II) OR ALPHANUMERIC(AA). MUST AGREE WITH THE TYPE OF THE VARIABLE WITHIN THE PROGRAM.
RR,II OR AA	NEW VALUE OF THE VARIABLE AVAR TO BE SUPPLIED IN THE APPROPRIATE FIELD DETERMINED BY THE VALUE OF ITYPE.
RROLD,IIOLD OR AAOLD	THE PREVIOUS VALUE OF THE VARIABLE AVAR IN THE APPROPRIATE FIELD ACCORDING TO THE ITYPE VALUE. INTEGER OR ALPHANUMERIC DATA WILL BE TESTED EXACTLY, REAL DATA TO 5 SIGNIFICANT DIGITS. IF THE CURRENT VALUE IS DIFFERENT, THE PROGRAM WILL TERMINATE WITH AN ERROR MESSAGE. IF ZERO OR BLANK IS SUPPLIED,NO CHECK IS PERFORMED.

THESE A.2 CARDS WILL BE PROCESSED UNTIL A BLANK VALUE FOR AVAR IS ENCOUNTERED. NO FURTHER INPUT IS REQUIRED.

CARD A.3	FORMAT (3A4, 4F12.0)
UNITL	UNIT OF LENGTH (4 CHARACTERS)
UNITM	UNIT OF FORCE (MASS) (4 CHARACTERS)
UNITT	UNIT OF TIME (4 CHARACTERS).

NOTE : UNITL, UNITM AND UNITT SHOULD CORRESPOND TO THE USER'S INPUTS. THROUGHOUT THIS DESCRIPTION, INCHES, POUNDS AND SECONDS (IN,LBS,SEC) ARE USED AS SAMPLE UNITS.

GRAVTY(I),I=1,3	THE X, Y AND Z COMPONENTS (IN/SEC**2) OF THE GRAVITY VECTOR. NORMALLY THIS IS USED AS THE GRAVITY FORCE VECTOR ACTING ON THE SEGMENTS. THIS VECTOR DEFINES THE INERTIAL OR GROUND REFERENCE COORDINATE SYSTEM TO BE USED BY THE PROGRAM. THE ORIENTATION OF OTHER COORDINATE REFERENCE SYSTEMS (E.G., VEHICLE AND LOCAL SEGMENT) ARE DEFINED LATER WITH RESPECT TO THIS INERTIAL REFERENCE COORDINATE SYSTEM. ONE CAN THEREFORE DEFINE ANY DESIRED COORDINATE SYSTEMS TO MEET INDIVIDUAL REQUIREMENTS.
G	THE VALUE OF G (IN/SEC**2). IF BLANK OR ZERO, THE MAGNITUDE OF THE GRAVITY VECTOR WILL BE USED. SUPPLYING THE VALUE OF G PERMITS ONE TO SPECIFY A DIFFERENT GRAVITY VECTOR ABOVE (E.G., ZERO) FOR SPECIAL APPLICATIONS.

CARD A.4	FORMAT (2I4, 4F8.0)
NDINT	NUMBER OF ITERATIONS FOR FINAL CONVERGENCE TEST OF THE INTEGRATOR SUBROUTINE DINT (MINIMUM VALUE = 2, SUGGESTED VALUE = 4).
NSTEPS	NUMBER OF INTEGRATION STEPS (OR OUTPUT TIME POINTS) FOR THE INTEGRATOR ROUTINE. MAY BE ZERO TO OBTAIN INITIAL CONDITIONS.
DT	MAIN PROGRAM TIME INTERVAL FOR INTEGRATOR ROUTINE OUTPUT (SEC). TOTAL TIME OF RUN WILL BE NSTEPS*DT SECONDS WITH MAIN PROGRAM TAPE 1, PRINTER PLOT AND OPTIONAL OUTPUT PRODUCED EVERY DT SECONDS.
HO	INITIAL INTEGRATOR STEP SIZE (SEC).
HMAX	MAXIMUM INTEGRATOR STEP SIZE (SEC). FOR BEST EFFICIENCY DT SHOULD BE AN INTEGRAL MULTIPLE OF HMAX AND HMAX A POWER OF TWO MULTIPLE OF HO. (SUGGESTED VALUE = 0.001 SEC.)
HMIN	MINIMUM INTEGRATOR STEP SIZE (SEC). IF A FIXED STEP SIZE IS DESIRED, SET HMIN GREATER THAN HMAX, AND STEP SIZE WILL DOUBLE FROM HO UNTIL HMAX IS ACHIEVED.

CARD A.5 FORMAT (36I2)

NPRT(I),I=1,36 AN ARRAY OF INDICATORS THAT CONTROL VARIOUS OPTIONAL OUTPUT FEATURES OF THE PROGRAM. GENERALLY, A BLANK OR ZERO VALUE INDICATES NO OUTPUT FOR THAT ITEM AND A VALUE OF ONE WILL PRODUCE OUTPUT EACH TIME THE ROUTINE IS EXECUTED. THE PRINTED OUTPUT PRODUCED BY ELEMENTS 7-27 IS INTENDED FOR DIAGNOSTIC OR "CHECK OUT" PURPOSES ONLY, CAN PRODUCE LARGE AMOUNTS OF OUTPUT AND SHOULD NOT BE USED FOR LONG OR PRODUCTION RUNS. IT IS NOT COMPLETELY LABELED AND ONE SHOULD CONSULT THE LISTING OF THE SUBROUTINE FOR A DESCRIPTION OF THE ITEMS THAT ARE PRINTED.

THE NPRT ARRAY (* - SEE NOTES BELOW)

ELEMENT NO.	SUBROUTINE	OUTPUT PRODUCED
1 (1*)	MAIN	OUTPUT UNIT NO. 1
2 (1*)	MAIN	SUBROUTINE ELTIME TABLE
3 (1*)	MAIN	SUBROUTINE PRINT OUTPUT
4 (3*)	OUTPUT,POSTPR	OUTPUT UNIT NO. 8, PLOTS
5 (1*)	PRIPLT	Y-Z VIEW PRINTER PLOTS
6 (1*)	PRIPLT	X-Z VIEW PRINTER PLOTS
7	BINPUT	HA AND HB ARRAYS
8 (2*)	DAUX	IJK, RHS AND C ARRAYS
9	DAUX	SUBROUTINE PRINT OUTPUT
10	IMPULS	DIAGNOSTIC OUTPUT
11	SETUP1	U2,V1 ARRAYS
12	VISPR	DIAGNOSTIC OUTPUT
13	PRIPLT	CJOINT ARRAY
14	WINDY	WIND FORCES
15	BELTG	DIAGNOSTIC OUTPUT
16	HBELT	HARNESS-BELT FORCES
17	EDEPTH	DIAGNOSTIC OUTPUT
18	NOT USED	
19	NOT USED	
20	CHAIN	SEGLP,SEGLV
21	AIRBAG	DIAGNOSTIC OUTPUT
22	AIRBG1	DIAGNOSTIC OUTPUT
23	NOT USED	
24	UPDATE	ROLL-SLIDE TEST OUTPUT
25	DINT	CONVERGENCE TEST DATA
26 (4*)	DINT,POSTPR	TABULAR TIME HISTORY OUTPUT
27	EQUILB	INTERMEDIATE RESULTS
28 (5*)	HPTURB	HARNESS BELT FORCES

NOTES CONCERNING ELEMENTS OF THE NPRT ARRAY

1* FOR ELEMENTS 1,2,3,5 AND 6, THE VALUE INDICATES THE FREQUENCY, ZERO FOR NO OUTPUT (FOR ELEMENT NO. 2, THE ELTIME TABLE WILL BE PRINTED ONCE AT THE END OF THE RUN), AND A NON-ZERO POSITIVE VALUE (N) WILL PRODUCE OUTPUT EVERY N*DT (FROM CARD A.4) SECONDS.

2* A VALUE OF NPRT(8) = 2 WILL PRINT THE DESIGNATED ARRAYS BEFORE AND AFTER THE FIRST CALL TO SUBROUTINE FSMSOL ONLY.

3* THE VALUE OF NPRT(4) IS USED (AFTER VERSION 13A) TO CONTROL

(1) WRITE THE TABULAR TIME HISTORIES (SPECIFIED BY CARDS H AND THE ALLOWED CONTACTS ON CARDS F) ON EITHER
  (A) THE MULTIPLE OUTPUT UNITS (NO. 21 AND UP) BY SUBROUTINE OUTPUT, OR
  (B) THE PRIMARY OUTPUT UNIT (NO. 6) BY SUBROUTINE HEDING.

(2) STORE THE TIME HISTORY DATA ON OUTPUT UNIT NO. 8 BY SUBROUTINE OUTPUT TO BE LATER USED BY SUBROUTINE POSTPR.

(3) GENERATE PLOTS OF THE TIME HISTORY DATA (SPECIFIED ON CARDS I) BY SUBROUTINE POSTPR.

THE PERMISSIBLE VALUES OF NPRT(4) RANGE FROM -3 TO +4 AS FOLLOWS:

	SUPPLIED VALUE FOR NPRT(4)							
	+4	+3	+2	+1	0	-1	-2	-3
1 CONTROL CARDS								
MULTIPLE OUTPUT UNITS	YES	NO	NO	YES	YES	NO	NO	NO
OUTPUT UNIT NO. 8	YES	YES	YES	YES	NO	YES	YES	YES
2 CARD INPUT								
CARDS B.1-H.7	YES	YES	YES	YES	YES	NO	NO	NO
CARD H.8	NO	YES	YES	YES	NO	YES	YES	YES
CARDS I	NO	YES	NO	YES	NO	YES	NO	YES
3 MAIN PROGRAM OPERATION								
INTEGRATE AND/OR RESTART	YES	YES	YES	YES	YES	NO	NO	NO
CALL SUBROUTINE POSTPR	NO	YES	YES	YES	NO	YES	YES	YES
4 PRINT TIME HISTORIES								
MULTIPLE OUTPUT UNITS	YES	NO	NO	YES	YES	NO	NO	NO
PRIMARY OUTPUT UNIT	NO	YES	YES	NO	NO	NO	YES	YES
5 OUTPUT UNIT NO. 8								
WRITE (SUB OUTPUT)	YES	YES	YES	YES	NO	NO	NO	NO
READ (SUB POSTPR)	NO	YES	YES	YES	NO	YES	YES	YES
6 GENERATE PLOTS (CARDS I)	NO	YES	NO	YES	NO	YES	NO	YES

4* NPRT(26) CONTROLS THE FREQUENCY OF THE TABULAR TIME HISTORY OUTPUT. VALUES OF 0,1 OR 2 ARE PERMISSIBLE TO CONTROL

(A) IF THE TABULAR TIME HISTORIES ARE PRINTED ON THE MULTIPLE OUTPUT UNITS 21 AND UP (NPRT(4) = 0,1 OR 4), A VALUE OF NPRT(26) = 0 OR 1 WILL PRINT AT THE END OF EACH SUCCESSFUL INTEGRATION STEP. A VALUE OF NPRT(26) = 2 WILL PRINT AT EACH INTERMEDIATE TIME POINT OF EACH INTEGRATION STEP.

(B) IF OUTPUT UNIT NO. 8 IS GENERATED (NPRT(4) > 0), RECORDS ARE WRITTEN AT THE SAME FREQUENCY SPECIFIED IN (A) ABOVE.

(C) IF THE TABULAR TIME HISTORIES ARE PRINTED FROM OUTPUT UNIT NO. 8 (NPRT(4) = +2,+3,-2 OR -3), A VALUE OF NPRT(26) EQUAL TO
0 WILL PRINT ONE LINE EVERY DT (FROM CARD A.4) SECONDS;
1 WILL PRINT AT THE END OF EACH SUCCESSFUL INTEGRATION STEP;
2 WILL PRINT AT EVERY INTERMEDIATE TIME POINT OF EACH STEP.

5* NPRT(28) CONTROLS THE FREQUENCY AND LEVEL OF DIAGNOSTIC HARNESS BELT FORCES OUTPUT PRODUCED. VALUES OF 0,1,2 AND 3 ARE ALLOWED AS FOLLOWS: (EACH VALUE INCLUDES OUTPUT OF ALL LOWER VALUES)

(0) - PRODUCES A TABLE OF THE FINAL HARNESS BELT FORCES AT EACH POINT IN PLAY AT THE SAME TIME POINTS AS OUTPUT IS PRODUCED BY SUBROUTINE PRINT AS SPECIFIED BY NPRT(3).

(1) - PRINTS A TABLE OF THE FINAL HARNESS BELT FORCES AT EACH POINT IN PLAY AT EACH TIME POINT OF SUBROUTINE HPTURB.

(2) - PRINTS A TABLE OF THE HARNESS BELT FORCES AT EACH POINT IN PLAY FOR EVERY ITERATION STEP OF SUBROUTINE HPTURB.

(3) - PRINTS THE RHS,IJK AND C ARRAYS BEFORE THE CALL TO FSMSOL AT EACH ITERATION STEP AT EACH TIME POINT OF HPTURB

IF NPRT(4) IS NEGATIVE, INPUT CARDS B.1-H.7 SHOULD NOT BE SUPPLIED. *

B. SUBROUTINE BINPUT

CARD B.1 FORMAT (2I6, 8X, 5A4)

NSEG	THE NUMBER OF SEGMENTS FOR THE CRASH VICTIM. THE MAXIMUM VALUE IS 30, BUT THIS INCLUDES ONE FOR THE GROUND, NBAG AIRBAGS, AND THE NEW SEGMENTS (INCLUDING THE PRIMARY VEHICLE) FOR WHICH PRESCRIBED MOTION IS DEFINED ON CARDS C.
NJNT	THE NUMBER OF JOINTS (MAXIMUM = 29). NOTE: NORMALLY NJNT = NSEG-1, BUT JOINT NUMBERS NVEH-1 AND NGRND-1 MAY BE USED TO CONNECT THE VEHICLE AND THE GROUND TO A LOWER NUMBERED SEGMENTS.
BDYTTL(I),I=1,5	DESCRIPTION OF THE CRASH VICTIM (20 CHARACTERS).

CARDS B.2.A - B.2.I (NSEG CARDS) FORMAT (A4, 1X, A1, 10F6.0)

EACH CARD (I) FOR I = 1, NSEG WILL CONTAIN INPUT DATA FOR THE ITH SEGMENT. THE SEGMENT IDENTIFYING NUMBERS (I) WILL BE REFERRED TO ON LATER INPUT CARDS.

SEG(I)	AN ABBREVIATION OF THE NOMENCLATURE OF THE ITH SEGMENT (4 CHARACTERS).
CGS(I)	THE PLOT SYMBOL OF THE SEGMENT C.G. (1 CHARACTER).
W(I)	THE WEIGHT OF THE SEGMENT (LBS).
PHI(J,I),J=1,3	THE PRINCIPAL MOMENTS OF INERTIA OF THE SEGMENT ABOUT THE X, Y, AND Z AXES OF THE SEGMENT (LBS-SEC**2-IN). THERE ARE NO RESTRICTIONS ON THE VALUES OF W(I) OR PHI(J,I), THEY MAY BE NEGATIVE OR ZERO. IF ANY COMPONENT IS ZERO, IT IS ASSUMED THAT THE SYSTEM IS SUITABLY CONSTRAINED SO THAT THE SYSTEM MATRIX IS NON-SINGULAR.
BD(J,I),J=1,3	THE X, Y, AND Z SEMIAXES OF THE SEGMENT CONTACT ELLIPSOID (IN).
BD(J,I),J=4,6	THE LOCATION OF THE CENTER OF THE SEGMENT CONTACT ELLIPSOID, WITH RESPECT TO THE CENTER OF GRAVITY OF THE SEGMENT, IN THE LOCAL BODY SEGMENT REFERENCE(IN). THESE PRIMARY CONTACT ELLIPSOIDS ARE GIVEN THE SAME INDENTIFYING NUMBER AS THE SEGMENT. THEY MAY BE REDEFINED WITH AN ARBITRARY ORIENTATION ON CARDS D.5.

IN NJNT IS ZERO ON CARD B.1, CARDS B.3 - B.5 ARE NOT REQUIRED.

CARDS B.3.A1 - B.3.J1   FORMAT (A4, 1X, A1, 2I4, 6F6.0)
(NJNT SETS OF CARDS, 2 CARDS PER SET. THE FIRST CARD OF EACH SET IS DESCRIBED ON THIS PAGE, THE SECOND CARD ON THE NEXT PAGE.)

EACH CARD (J) FOR J = 1, NJNT WILL CONTAIN INPUT DATA FOR THE JTH JOINT. THE JOINT IDENTIFYING NUMBERS (J) WILL BE REFERRED TO ON LATER INPUT CARDS.

JOINT(J)	AN ABBREVIATION OF THE NOMENCLATURE OF THE JTH JOINT (4 CHARACTERS).
JS(J)	PLOT SYMBOL OF THE JOINT LOCATION (1 CHARACTER).
JNT(J)	MAGNITUDE INDICATES THE NUMBER OF THE SEGMENT THAT IS CONNECTED TO SEGMENT J+1 BY JOINT J. IF NEGATIVE, JOINT J IS ASSOCIATED WITH A FLEXIBLE ELEMENT. IF ZERO, SEGMENT J+1 IS THE REFERENCE SEGMENT OF ANOTHER BODY. (|JNT(J)| < J+1).
IPIN(J)	0 - THERE ARE TO BE NO CONSTRAINTS ON JOINT J. 1 - JOINT J IS PINNED (HINGE). 2 - JOINT J IS NOT PINNED (BALL AND SOCKET). 3 - JOINT J IS GLOBALGRAPHIC (BALL AND SOCKET) 4 - JOINT J IS AN EULER JOINT. NON-ZERO VALUES FOR IPIN MAY BE SUPPLIED AS POSITIVE OR NEGATIVE TO INDICATE THAT THE INITIAL CONDITION OF THE JOINT IS UNLOCKED (POSITIVE) OR LOCKED (NEGATIVE). AN EULER JOINT MAY USE THE GLOBALGRAPHIC OPTION. SPECIFY IGLOB = 1 ON CARD F.4.A THE INITIAL STATE OF AN EULER JOINT IS SET BY USE OF IPIN AS FOLLOWS

IPIN	IEULER	STATE
4	8	FREE
- 4	7	ALL AXES LOCKED
- 5	6	SPIN FREE, OTHERS LOCKED
- 6	5	NUTATION FREE, OTHERS LOCKED
- 7	4	PRECESSION FREE, OTHERS LOCKED
- 8	3	SPIN LOCKED, OTHERS FREE
- 9	2	NUTATION LOCKED, OTHERS FREE
-10	1	PRECESSION LOCKED, OTHERS FREE

WHERE PRECESSION IS ABOUT THE Z AXIS OF THE JOINT REFERENCE (YPR1) IN SEGMENT NO. JNT(J), NOTATION ABOUT THE RESULTANT X AXIS, AND SPIN ABOUT THE RESULTANT Z AXIS OF THE JOINT REFERENCE (YPR2) IN SEGMENT NO. J+1.
IF IPIN IS LESS THAN -3 PROGRAM WILL SET IEULER AS ABOVE AND THEN SET IPIN = -4.

SR(I,2*J-1),I=1,3	COORDINATES OF LOCATION OF JOINT J (IN.) IN THE LOCAL REFERENCE SYSTEM OF SEGMENT JNT(J).
SR(I,2*J),I=1,3	COORDINATES OF LOCATION OF JOINT J (IN.) IN THE LOCAL REFERENCE SYSTEM OF SEGMENT J+1.

```
CARDS B.3.A2 - B.3.J2   FORMAT (14X, 9F6.0, 6I2) **                      *
(ONE OF THESE CARDS MUST FOLLOW EACH CARD FROM PREVIOUS PAGE.)            |

    YPR1(I,J),I=1,3       THE ROTATION ANGLES (DEGREES) ABOUT THE Z, Y   |
                          AND X AXES, RESPECTIVELY, OF THE LOCAL REF-    |
                          ERENCE OF SEGMENT NO. JNT(J) TO SPECIFY THE    |
                          PRINCIPAL AXES OF JOINT J. THE ORDER OF THESE  *
                          ROTATIONS IS SPECIFIED BY ID1 BELOW.           *

    YPR2(I,J),I=1,3       THE ROTATION ANGLES (DEGREES) ABOUT THE Z, Y   |
                          AND X AXES, RESPECTIVELY, OF THE LOCAL REF-    |
                          ERENCE OF SEGMENT NO. J+1 TO SPECIFY THE       |
                          PRINCIPAL AXES OF JOINT J. THE ORDER OF THESE  *
                          ROTATIONS IS SPECIFIED BY ID2 BELOW.           *
                          THE Z AXIS IS THE REFERENCE AXIS TO DEFINE
                          FLEXURE. THE Y AXIS IS USED AS THE PIN AXIS
                          EXCEPT FOR THE SPECIAL EULER JOINTS. THE XY
                          PLANE IS USED FOR GLOBALGRAPHIC JOINTS WITH
                          X AS THE REFERENCE AXIS.

    YPR3(I,J),I=1,3       THE CENTER OF SYMMETRY (DEGREES) FOR EULER     *
                          JOINTS (USED ONLY IF |IPIN(J)| = 4) SUPPLIED IN |
                          THE ORDER PRECESSION, NOTATION AND SPIN. JOINT |
                          TORQUES FOR EULER JOINTS ARE A FUNCTION OF     *
                          THE DEVIATION OF THE EULER ANGLES FROM THESE   *
                          ANGLES. PREVIOUS VERSIONS (BEFORE 19A) OF      *
                          PROGRAM ASSUMED VALUES OF ZERO.                *

    ID1(I,J),I=1,3        VALUES OF 1,2 AND 3, CORRESPONDING TO THE X,   *
                          Y AND Z AXES, SPECIFYING THE ORDER OF THE AXES *
                          ABOUT WHICH THE ROTATIONS GIVEN IN YPR1 ARE    *
                          TO BE PERFORMED. ZERO OR BLANK VALUES WILL     *
                          DEFAULT TO THE ORDER 3,2 AND 1 TO SPECIFY THE  *
                          NORMAL YAW, PITCH AND ROLL SEQUENCE, I.E.,     *
                                                                         *
                            YAW ABOUT ORIGINAL  Z AXIS USING YPR1(1,J).  *
                          PITCH ABOUT RESULTANT Y AXIS USING YPR1(2,J).  *
                           ROLL ABOUT RESULTANT X AXIS USING YPR1(3,J).  *
                                                                         *
                          THE SAME AXIS CANNOT BE SPECIFIED FOR TWO OR   *
                          MORE CONSECUTIVE ROTATIONS. HOWEVER, THE THIRD *
                          AXIS MAY BE THE SAME AS THE FIRST, PROVIDED IT *
                          IS SUPPLIED AS A NEGATIVE NUMBER, IN WHICH CASE *
                          THE UNUSED VALUE OF YPR1 WILL BE USED ABOUT    *
                          THE INDICATED AXIS. E.G., VALUES OF 3,1 AND -3 *
                          WILL SPECIFY THE NORMAL EULER ROTATIONS WHERE  *
                          YPR1 IS SUPPLIED IN THE ORDER PRECESSION,      *
                          SPIN AND NOTATION TO COMPUTE                   *
                                                                         *
                          PRECESSION (YPR1(1,J)) ABOUT ORIGINAL Z AXIS.  *
                          NUTATION (YPR1(3,J)) ABOUT RESULTANT X AXIS,   *
                          AND SPIN (YPR1(2,J)) ABOUT RESULTANT Z AXIS.   *

    ID2(I,J),I=1,3        SPECIFIES THE ORDER OF THE ROTATIONS GIVEN BY  *
                          YPR2 IDENTICAL TO THE DESCRIPTION OF ID1.      *
```

```
CARDS B.4.A - B.4.J      FORMAT (2 (4F6.0, F12.0))
    (NJNT SETS OF CARDS, ONE FOR EACH JOINT J. IF |IPIN(J)| # 4,
     EACH SET READS VALUES FOR 3*J-2 AND 3*J-1 ON ONE CARD ONLY.
    IF |IPIN(J)| = 4 , JOINT J IS AN EULER JOINT AND A SECOND CARD
    IS NECESSARY TO READ VALUES FOR 3*J)

    SPRING(I,3*J-2),         THE FLEXURAL SPRING CHARACTERISTICS FOR
           I=1,5             JOINT J. IF J IS AN EULER JOINT, THE SPRING
                             CHARACTERISTICS ABOUT THE PRECESSION AXIS.
                             IF JOINTF(J) # 0 (ON CARD F.5.A), THESE       |
                             VALUES ARE NOT USED AND SHOULD BE ZERO.       |

    SPRING(I,3*J-1),         THE TORSIONAL SPRING CHARACTERISTICS FOR
           I=1,5             JOINT J. IF J IS AN EULER JOINT, THE SPRING
                             CHARACTERISTICS ABOUT THE NUTATION AXIS.

    SPRING(I,3*J),           SECOND CARD OF EACH SET IS REQUIRED
           I=1,5             ONLY IF J IS AN EULER JOINT, THE SPRING
                             CHARACTERISTICS ABOUT THE SPIN AXIS.

               I=1           LINEAR SPRING COEFFICIENT
                             (IN-LBS/DEG).

               I=2           QUADRATIC SPRING COEFFICIENT
                             (IN-LBS/DEG**2).

               I=3           CUBIC SPRING COEFFICIENT
                             (IN-LBS/DEG**3).

               I=4           ENERGY DISSIPATION COEFFICIENT
                             (DIMENSIONLESS).
                             A VALUE OF 1. SPECIFIES NO LOSS
                             A VALUE OF 0. SPECIFIES MAXIMUM LOSS

               I=5           JOINT STOP LOCATION WITH RESPECT TO
                             THE CENTER OF SYMMETRY (DEG).
                             FOR A VALUE OF ZERO THE ROUTINE WILL USE ONLY
                             THE LINEAR SPRING COEFFICIENT AND WILL APPLY
                             THE ENERGY DISSIPATION FACTOR

    ANG(I,J),I=1,3           THE APPROXIMATE INITIAL ROTATION ANGLES,      *
                             IN THE ORDER PRECESSION, NUTATION AND SPIN,   *
                             (DEGREES) FOR JOINT J WHICH IS AN EULER JOINT. *
                             THESE ARE USED AS THE INITIAL ANGLES FOR THE  *
                             MEMORY MODE USED BY SUBROUTINE EULRAD AND     *
                             NEED NOT BE EXACT. THE VALUES ARE ABSOLUTE    *
                             AND NOT RELATIVE TO THE CENTER OF SYMMETRY.   *
```

CARDS B.5.A - B.5.J   FORMAT (5F6.0, 18X, 2F6.0)
(NJNT SETS OF CARDS, ONE FOR EACH JOINT J. IF |IPIN(J)| # 4, VALUES FOR 3*J-2 ARE ON ONE CARD ONLY. IF |IPIN(J)| = 4, J IS AN EULER JOINT AND VALUES FOR 3*J-1 AND 3*J ARE REQUIRED ON A SECOND AND THIRD CARD OF EACH SET.)

VISC(I,3*J-2), I=1,7 — THE VISCOUS CHARACTERISTICS FOR JOINT J. IF J IS AN EULER JOINT, THE VISCOUS CHARACTERISTICS ABOUT THE PRECESSION AXIS.

VISC(I,3*J-1), I=1,7 — THE SECOND CARD OF EACH SET IS REQUIRED ONLY IF J IS AN EULER JOINT. THE VISCOUS CHARACTERISTICS ABOUT THE NUTATION AXIS.

VISC(I,3*J) I=1,7 — THE THIRD CARD OF EACH SET IS REQUIRED ONLY IF J IS AN EULER JOINT. THE VISCOUS CHARACTERISTICS ABOUT THE SPIN AXIS.

I=1 — VISCOUS COEFFICIENT (IN-LB-SEC/DEG).

I=2 — COULOMB FRICTION COEFFICIENT (IN-LB).

I=3 — RELATIVE ANGULAR VELOCITY OF JOINT AT WHICH FULL COULOMB FRICTION IS APPLIED (DEG/SEC). MUST BE GREATER THAN 0.

I=4 — T1: THE MAXIMUM TORQUE (IN-LBS) ALLOWED FOR A LOCKED JOINT (OR EULER AXIS). IF EXCEEDED,THE JOINT WILL UNLOCK. IF T1 = 0, THE TEST WILL NOT BE PERFORMED. NOTE: IF JOINT J IS LOCKED, IF T1=0, AND IF SUBROUTINE EQUILB IS CALLED, THEN VISC(4,3*J-2) WILL BE SET BY SUBROUTINE EQUILB. (SEE DESCRIPTION UNDER CARDS G.6)

I=5 — T2: THE MINIMUM TORQUE (IN-LBS) ALLOWED FOR JOINT J TO REMAIN UNLOCKED. IF T2 = 0, THE TEST WILL NOT BE PERFORMED.

I=6 — T3: THE MINIMUM ANGULAR VELOCITY (RAD/SEC) NECESSARY FOR JOINT J TO REMAIN UNLOCKED. IF T3 = 0, THE TEST WILL NOT BE PERFORMED.

I=7 — E = (1+U)/2 WHERE U IS THE CLASSICAL COEFFICIENT OF RESTITUTION TO BE USED FOR THE IMPULSE OPTION IF THE JOINT HITS THE JOINT STOP (0<E<1 OR -1<U<+1). A VALUE OF E = 0 MEANS THAT THE IMPULSE OPTION WILL NOT BE EXERCISED FOR THIS JOINT.

```
CARDS B.6.A - B.6.I      FORMAT (12F6.0)
     (NSEG CARDS)

    SGTEST(1,1,I)        MAGNITUDE TEST FOR THE ANGULAR VELOCITY
                         OF SEGMENT NO. I (RAD/SEC).

    SGTEST(2,1,I)        ABSOLUTE ERROR TEST FOR THE ANGULAR
                         VELOCITY OF SEGMENT NO. I (RAD/SEC).

    SGTEST(3,1,I)        RELATIVE ERROR TEST FOR THE ANGULAR
                         VELOCITY OF SEGMENT NO. I (DIMENSIONLESS).

    SGTEST(1,2,I)        SAME AS ABOVE, BUT FOR THE LINEAR
          (2,2,I)        VELOCITY OF SEGMENT NO. I (IN/SEC).
          (3,2,I)

    SGTEST(1,3,I)        SAME AS ABOVE, BUT FOR THE ANGULAR
          (2,3,I)        ACCELERATION OF SEGMENT NO. I (RAD/SEC**2).
          (3,3,I)

    SGTEST(1,4,I)        SAME AS ABOVE BUT FOR THE LINEAR
          (2,4,I)        ACCELERATION OF SEGMENT NO. I (IN/SEC**2).
          (3,4,I)
```

THESE CONVERGENCE TESTS ARE PERFORMED IN SUBROUTINE DINT ON THE RESULTANT OF THE DERIVATIVE VECTORS. THE LINEAR VELOCITIES AND ACCELERATIONS ARE COMPUTED ONLY FOR REFERENCE SEGMENTS (I.E. SEGMENT NO. 1 AND THOSE SEGMENTS I WHERE JNT(I-1) = 0), THEREFORE ANY TEST NUMBERS SUPPLIED FOR LINEAR VELOCITIES AND ACCELERATIONS OF OTHER SEGMENTS WILL BE IGNORED. THE TESTS FOR CONVERGENCE ARE PERFORMED IN THE FOLLOWING ORDER :

1) IF THE MAGNITUDE TEST IS ZERO, NO TESTING IS DONE FOR THAT VARIABLE.

2) IF THE MAGNITUDE OF THE RESULTANT VECTOR IS LESS THAN THE MAGNITUDE TEST,THE ROUTINE HAS PASSED THE CONVERGENCE TEST FOR THAT VARIABLE.

3) IF THE ABSOLUTE ERROR TEST IS GREATER THAN ZERO, AND THE MAGNITUDE OF THE ABSOLUTE ERROR (DIFFERENCE BETWEEN THE PREDICTED AND COMPUTED VECTOR) IS LESS THAN THE ABSOLUTE ERROR TEST, THE ROUTINE HAS PASSED THE CONVERGENCE TEST FOR THAT VARIABLE.

4) IF THE RELATIVE ERROR OF THE MAGNITUDE OF THE ABSOLUTE ERROR COMPARED TO THE MAGNITUDE OF THE COMPUTED VECTOR IS GREATER THAN THE RELATIVE ERROR TEST, THE CONVERGENCE TEST HAS FAILED.

IF NFLX ≠ 0, CARDS B.7 ARE REQUIRED. EACH FLEXIBLE ELEMENT AS DEFINED ON CARDS B.3 CONTAINS AT LEAST THREE CONNECTED SEGMENTS CONSISTING OF A REFERENCE SEGMENT, ONE OR MORE INTERIOR SEGMENTS AND A TERMINATING SEGMENT. EACH JOINT IN THE ELEMENT SHOULD HAVE A NEGATIVE VALUE FOR JNT, AND THE NUMBER OF INTERIOR SEGMENTS WILL BE ONE LESS THAN THE NUMBER OF NEGATIVE VALUES OF JNT FOR EACH ELEMENT. NFLX IS THE TOTAL NUMBER OF INTERIOR SEGMENTS OF ALL FLEXIBLE ELEMENTS.

CARD B.7.A	FORMAT (18I4)
NFX	THE NUMBER OF INTERIOR SEGMENTS FOR WHICH HF ARRAYS ARE TO BE SUPPLIED.
KNT(K),K=1,NFX	THE INTERIOR SEGMENT IDENTIFICATION NUMBERS IN THE ORDER OF THE HF ARRAYS TO BE SUPPLIED. IF THE VALUES OF NFX AND KNT ARE NOT CONSISTENT WITH THE NEGATIVE VALUES OF JNT ON CARDS B.3 THE PROGRAM WILL TERMINATE WITH AN APPROPRIATE ERROR MESSAGE.

CARDS B.7.B - B.7.N	FORMAT (12F6.0)

(4*NFX CARDS, 4 CARDS FOR EACH SEGMENT IN THE ORDER AS THEY ARE DEFINED IN THE KNT VECTOR.)

(HF(I,J,K),J=1,12) ,I=1,4	THE COEFFICIENTS OF THE QUADRATIC FORM FUNCTION USED TO DEFINE THE ORIENTATION OF INTERIOR SEGMENT KNT(K) WITH RESPECT TO REFERENCE SEGMENT OF THE ELEMENT.

FORM THE COLUMN VECTOR V WITH FOUR COMPONENTS Y,P,R AND 1, WHERE Y,P,R ARE THE YAW, PITCH AND ROLL OF THE TERMINATING SEGMENT RELATIVE TO THE REFERENCE SEGMENT. LET H BE A SYMMETRIC 4X4 MATRIX SUCH THAT F(V) = 1/2 V.HV REPRESENTS A QUADRATIC SCALAR FUNCTION OF THE VARIABLES Y,P AND R IN RADIANS. THUS

```
  YAW OF SEGMENT KNT(K) = 1/2 V.HF(I,J  ,K)V
PITCH OF SEGMENT KNT(K) = 1/2 V.HF(I,J+4,K)V
 ROLL OF SEGMENT KNT(K) = 1/2 V.HF(I,J+8,K)V     (I,J=1,4)
```

C. SUBROUTINE VINPUT

THESE C CARDS ARE USED TO PRESCRIBE THE MOTION (ACCELERATION TIME HISTORY) OF SPECIFIED SEGMENTS. NORMALLY ONLY ONE SET IS SUPPLIED WITH MSEG (LAST ITEM ON CARD C.2) EQUAL TO ZERO (OR BLANK) TO PRESCRIBE THE MOTION OF THE PRIMARY VEHICLE (SEGMENT NO. NSEG+1). HOWEVER, MULTIPLE SETS MAY BE SUPPLIED (MAXIMUM = 6) WITH MSEG = 0 ON THE LAST SET TO DENOTE THE PRIMARY VEHICLE.

SEVERAL OPTIONS ARE AVAILABLE FOR EACH PRESCRIBED MOTION. THE REQUIRED INPUTS FOR EACH OPTION ARE AS FOLLOWS:

OPTION 1: HALF SINE WAVE DECELERATION IMPULSE (NATAB = 0)

REQUIRED INPUTS - CARD C.1; CARD C.2.A: ANGLE(1), ANGLE(2), VIPS, VTIME, XO, NATAB=0, MSEG.

OPTION 2: TABULAR UNIDIRECTIONAL DECELERATION (NATAB > 0)

REQUIRED INPUTS - CARD C.1; CARD C.2.A: ANGLE(1), ANGLE(2), VIPS, XO, NATAB>0, ATO, ADT, MSEG; CARDS C.3.

OPTION 3: SIX DEGREE OF FREEDOM DECELERATION (NATAB < 0 AND LTYPE = 0)

REQUIRED INPUTS - CARD C.1; CARD C.2.A: ANGLE(1), ANGLE(2), ANGLE(3), VIPS, XO, NATAB<0, ATO, ADT, MSEG; CARD C.2.B: LTYPE=0, VMEG; CARDS C.4.

OPTION 4: SPLINE FIT POSITION, VELOCITY OR ACCELERATION DATA (NATAB <0 AND LTYPE > 0)

REQUIRED INPUTS - CARD C.1; CARD C.2.A: NATAB<0, ATO, ADT, MSEG; CARD C.2.B; LTYPE>0, LFIT, NPTS; CARDS C.5.

THESE OPTIONS AND THEIR REQUIRED INPUTS HAVE BEEN ESTABLISHED IN SUCH A MANNER THAT ANY PREVIOUS INPUT DECKS ARE STILL ACCEPTABLE AS INPUT, EXCEPT THAT CARD C.2.B WAS ADDED FOR OPTION 3 FOR VERSION 18 OF THE CVS PROGRAM. FOR VERSION 19, CARD C.2.B HAS BEEN MODIFIED AND OPTION 4 (CARDS C.5) AND THE MULTIPLE PRESCRIBED MOTION WERE ADDED.

CARD C.1	FORMAT (20A4) **	
VPSTTL(I),I=1,20	DESCRIPTION OF THE CRASH VEHICLE DECELERATION (80 CHARACTERS).	

CARD C.2.A — FORMAT (8F6.0, I6, 2F6.0, I6)

ANGLE(I),I=1,3	OPTIONS 1 AND 2: ANGLE(1) AND ANGLE(2) (DEG) ARE THE AZIMUTH AND ELEVATION (OBLIQUE ANGLES) OF THE DIRECTION OF THE DECELERATION IMPULSE. THE INITIAL YAW, PITCH AND ROLL OF THE VEHICLE ARE ASSUMED TO BE ZERO. OPTION 3: THE THREE ANGLES ARE THE INITIAL YAW, PITCH AND ROLL (DEG) OF THE VEHICLE.
VIPS	THE INITIAL VELOCITY (IN/SEC) OF THE VEHICLE. FOR OPTION 1, A NEGATIVE VALUE MAY BE SUPPLIED TO INDICATE THAT THE VEHICLE WILL ACCELERATE FROM AN INITIAL VELOCITY OF ZERO TO -VIPS.
VTIME	THE TIME DURATION (SEC) OF THE HALF SINE WAVE DECELERATION IMPULSE. CANNOT BE ZERO OR BLANK FOR OPTION 1.
XO(I),I=1,3	THE X, Y AND Z COORDINATES (IN) OF THE VEHICLE REFERENCE ORIGIN IN INERTIAL REFERENCE.
NATAB	NUMBER OF TIME POINTS OF VEHICLE DECELERATION DATA TO BE SUPPLIED OR GENERATED BY THE PROGRAM. THE ALGEBRAIC SIGN OF NATAB DETERMINES THE OPTION OF PRESCRIBED MOTION AS FOLLOWS: IF NATAB = 0 (OPTION 1), THE IMPULSE IS AN ANALYTICAL HALF SINE WAVE FUNCTION THAT (VIPS>0) DECELERATES THE VEHICLE FROM AN INITIAL VELOCITY OF VIPS TO ZERO, OR (VIPS<0) ACCELERATES THE VEHICLE FROM AN INITIAL VELOCITY OF ZERO TO -VIPS IN VTIME SEC. IF NATAB > 0 (OPTION 2), THE VEHICLE MOTION IS UNIDIRECTIONAL AND NATAB VALUES OF LINEAR DECELERATION ARE TO BE SUPPLIED ON CARDS C.3. NATAB SHOULD BE ODD, MAXIMUM VALUE IS 99. IF NATAB < 0 (OPTIONS 3 AND 4), THE PRESCRIBED MOTION IS SPECIFIED ON EITHER CARDS C.4 OR C.5. HERE MATAB (= -NATAB) IS THE NUMBER OF TIME POINTS OF ACCELERATION DATA TO BE SUPPLIED ON CARD C.4 OR COMPUTED FROM THE SPLINE FIT DATA ON CARDS C.5. MAXIMUM VALUE OF MATAB IS 101.
ATO,ATD	THE FIRST TIME AND FIXED TIME INTERVAL (SEC) FOR THE TABLE OF ACCELERATION DATA THAT FOR (OPTION 3) IS TO BE SUPPLIED ON CARDS C.4, OR (OPTION 4) IS TO BE COMPUTED FROM THE SPLINE FIT DATA TO BE SUPPLIED ON CARDS C.5.

MSEG	THE SEGMENT NUMBER ASSOCIATED WITH THIS PRE-SCRIBED DECELERATION TIME HISTORY. IF MSEG IS LESS THAN OR EQUAL TO NSEG (CARD B.1), THE MOTION OF SEGMENT NO. MSEG AS DEFINED ON CARDS B.2 WILL BE PRESCRIBED (NOTE: EXTREME CAUTION MUST BE EXERCISED IN USING THIS OPTION.) IF MSEG > NSEG, THE SETS MUST BE SUPPLIED IN THE ORDER MSEG=NSEG+1, NSEG+2, ETC., TO PRE-SCRIBE THE MOTION OF SECONDARY VEHICLE SEGMENTS. IF MSEG = 0, THIS IS THE LAST (OR ONLY) SET OF C CARDS TO BE SUPPLIED TO PRESCRIBE THE MOTION OF THE PRIMARY VEHICLE WHOSE SEGMENT NO. WILL BE ONE GREATER THAN NSEG OR THE LAST VALUE OF MSEG THAT WAS GREATER THAN NSEG.	*

CARD C.2.B FORMAT (3I6, 22X, 3F10.0) S

THIS CARD IS REQUIRED ONLY IF NATAB < 0 (OPTIONS 3 AND 4) S
NOTE: THIS CARD WAS ADDED FOR VERSION 18 OF THE CVS PROGRAM TO SUPPLY THE INITIAL ANGULAR VELOCITY AND REVISED FOR VERSION 19. A BLANK CARD SHOULD BE INSERTED HERE FOR ANY PREVIOUS INPUT DATA DECKS THAT UTILIZED THE SIX DEGREE OF FREEDOM OPTION ON CARDS C.4. S

LTYPE	OPTION 3: SUPPLY A VALUE OF ZERO OR BLANK FOR THE SIX DEGREE OF FREEDOM INPUT ON CARDS C.4.	S
	OPTION 4: A VALUE OF 1,2 OR 3 SPECIFIES THAT THE TABLES TO BE SUPPLIED ON CARDS C.5 ARE (1) POSITION, (2) VELOCITY OR (3) ACCELERATION DATA FOR EACH TIME POINT.	*
LFIT	THE DEGREE OF THE POLYNOMIALS TO BE SPLINE FITTED THROUGH THE TIME POINT DATA ON CARDS D.5. A VALUE OF 0, 1, 2 OR 3 MAY BE USED BUT THE DEGREE SHOULD BE SUFFICIENT TO PRODUCE CONT-INUITY FOR THE COMPUTED VELOCITY VALUES. FOR LTYPE = 1, SUPPLY LFIT = 2 OR 3. FOR LTYPE = 2, SUPPLY LFIT = 1,2 OR 3. FOR LTYPE = 3, SUPPLY LFIT = 0,1,2 OR 3. NOTE: FOR LFIT = 0, A CONSTANT VALUE IS ASSUMED FROM THE CURRENT TIME VALUE TO THE NEXT TIME VALUE BUT ROUND OFF ERRORS IN TIME COMPUTATIONS MAY NOT PRODUCE THE TIME DESIRED.	*
NPTS	THE NUMBER OF ACTUAL TIME POINT DATA TO BE SUPPLIED ON CARDS C.5.	*
VMEG(I),I=1,3	THE THREE COMPONENTS OF THE INITIAL ANGULAR VELOCITY (DEG/SEC) ABOUT THE LOCAL X, Y AND Z AXES OF THE VEHICLE.	S

```
CARDS C.3.A - C.3.N        FORMAT (12F6.0)

    THESE CARDS ARE REQUIRED ONLY IF NATAB > 0 (OPTION 2)

    DEC(I),I=1,NATAB       THE VALUES OF DECELERATION (G'S) OF THE VEHICLE
                           FOR THE NATAB EQUALLY SPACED TIME POINTS

                           T(I) = ATO + (I-1)*ADT  FOR I=1,NATAB.

                           SUPPLY 12 VALUES PER CARD, USE AS MANY CARDS
                           AS NECESSARY. SINCE A SIMPSON'S INTEGRATION
                           IS USED TO COMPUTE VELOCITY AND POSITION,
                           THE VALUE OF NATAB MUST BE ODD. THE LAST
                           VALUE, ATAB(1,NATAB) WILL BE USED TO INTEGRATE
                           FOR ANY TIME GREATER THAN T(NATAB-1).

CARDS C.4.A - C.4.M        FORMAT (10X, 6F10.0)

    THESE CARDS ARE REQUIRED IF NATAB<0 AND LTYPE=0 (OPTION 3)

    MATAB CARDS ARE REQUIRED WHERE MATAB = -NATAB. EACH CARD (I)
    WILL CONTAIN DATA FOR EQUALLY SPACED TIME POINTS T(I), WHERE

         T(I) = ATO + (I-1)*ADT   FOR I=1,MATAB.

    ATAB(J,I),J=1,3        THE X,Y AND Z COMPONENTS (G'S) OF THE LINEAR
                           DECELERATION OF THE VEHICLE ORIGIN AT TIME T(I).

    ATAB(J,I),J=4,6        THE ANGULAR ACCELERATIONS (DEG/SEC**2) ABOUT
                           THE LOCAL X,Y AND Z AXES OF THE VEHICLE AT T(I).

    NOTE: THE PROGRAM WILL INTEGRATE FOR VELOCITY AND POSITION BEYOND
    THE LAST TIME POINT USING THE VALUES AT THAT POINT. THE PROGRAM
    WILL PRINT AT INPUT TIME A COMPLETE TABLE OF THE INTEGRATED
    VELOCITY AND POSITION FROM THE SUPPLIED ACCELERATION DATA. THE
    INTEGRATION PROCEDURE IS NOT IDENTICAL TO THE PROGRAM INTEGRATOR.
```

```
CARDS C.5.A -C.5.M       FORMAT (7F10.0)                                   *

    THESE CARDS ARE REQUIRED IF NATAB<0 AND LTYPE>0 (OPTION 4)             *

    (LTYPE-1) CARDS ARE REQUIRED FIRST TO SET INITIAL CONDITIONS           *
    FOLLOWED BY NPTS CARDS CONTAINING TIME POINT DATA.                     *

IF LTYPE=1, THE INPUT TABLE IS POSITION DATA FOR NPTS TIME POINTS.         *

IF LTYPE=2, THE FIRST CARD IS THE INITIAL POSITION DATA, WHICH IS          *
FOLLOWED BY THE INPUT TABLE OF VELOCITY DATA FOR NPTS TIME POINTS.         *

IF LTYPE=3, THE FIRST CARD IS THE INITIAL POSITION DATA, THE SECOND        *
CARD IS THE INITIAL VELOCITY DATA, WHICH IS FOLLOWED BY THE INPUT          *
TABLE OF ACCELERATION DATA FOR NPTS TIME POINTS.                           *


    T(I)                 THE TIME (SEC) FOR THE DATA ON THIS CARD.         *
                         IF THIS CARD IS FOR INITIAL CONDITION DATA,       *
                         T(1) SHOULD BE ZERO OR BLANK, THE TIMES           *
                         SHOULD BE IN ASCENDING ORDER BUT DO NOT           *
                         HAVE TO BE EQUALLY SPACED.                        *

    XYZ(J,I),J=1,3       IF POSITION DATA, THE X,Y AND Z COORDINATES       *
                         (IN) OF THE VEHICLE ORIGIN IN THE INERTIAL        *
                         REFERENCE COORDINATE SYSTEM FOR TIME T(I).        *
                         IF VELOCITY DATA, THE X,Y AND Z COMPONENTS        *
                         (IN/SEC) OF VELOCITY OF THE VEHICLE ORIGIN        *
                         IN INERTIAL REFERENCE FOR TIME T(I).              *
                         IF ACCELERATION DATA, THE X,Y AND Z COMPONENTS    *
                         (IN/SEC**2) OF THE DECELERATION OF THE VEHICLE    *
                         ORIGIN IN INERTIAL REFERENCE FOR TIME T(I).       *

    XYZ(J,I),J=4,6       IF POSITION DATA, THE YAW, PITCH AND ROLL (DEG)   *
                         OF THE VEHICLE COORDINATE REFERENCE AXES WITH     *
                         RESPECT TO THE INERTIAL REFERENCE.                *
                         IF VELOCITY DATA, THE COMPONENTS OF ANGULAR       *
                         VELOCITY (DEG/SEC) ABOUT THE LOCAL X,Y,Z AXES.    *
                         IF ACCELERATION DATA, THE COMPONENTS OF ANGULAR   *
                         ACCELERATION (DEG/SEC**2) ABOUT THE LOCAL         *
                         X,Y AND Z AXES.                                   *

NOTE: THE PROGRAM WILL SPLINE FIT THE NPTS DATA POINTS FOR EACH OF THE     *
SIX COMPONENTS INDEPENDENTLY TO PRODUCE A PIECE-WISE SET OF POLYNOMIALS    *
OF DEGREE LFIT. THESE POLYNOMIALS ARE THEN EVALUATED TO PRODUCE A SET      *
OF ACCELERATION TABLES AT MATAB(= -NATAB) EQUALLY SPACED TIME POINTS       *
EQUIVALENT TO THE SIX DEGREE OF FREEDOM (OPTION 3) DATA OF CARDS C.4.      *
THE PROGRAM WILL THEN PRINT AT INPUT TIME A COMPLETE TABLE OF THE          *
INTEGRATED VELOCITY AND POSITION FROM THESE GENERATED ACCELERATION         *
DATA. THE INTEGRATION PROCEDURE USED IS NOT IDENTICAL TO THE PROGRAM       *
INTEGRATOR.                                                                *
```

D. SUBROUTINE SINPUT

CARD D.1	FORMAT (9I6)
NPL	THE NUMBER OF PLANES DESCRIBING CONTACT PANELS (30 MAXIMUM).
NBLT	THE NUMBER OF BELTS USED TO RESTRAIN THE CRASH VICTIM (8 MAXIMUM).
NBAG	THE NUMBER OF AIRBAGS USED TO RESTRAIN THE CRASH VICTIM (MAX = 5, MAX NSEG+NBAG = 20).
NELP	THE NUMBER OF CONTACT ELLIPSOIDS TO BE SUPPLIED ON CARDS D.5 (40 MAXIMUM).
NQ	THE NUMBER OF CONSTRAINTS TO BE SUPPLIED ON CARDS D.6. EACH CONSTRAINT TYPE 5 WILL BE CONSIDERED AS TWO CONSTRAINTS REQUIRING TWO SETS OF CARDS (NOTE: THE PROGRAM WILL LATER INCREMENT NQ BY 1 FOR EACH NF(1) = 0 ON CARDS F.1.B AND F.3.B AND THE FINAL MAXIMUM ON NQ IS 12).
NSD	THE NUMBER OF SPRING DAMPERS TO BE SUPPLIED ON CARDS D.8 (20 MAXIMUM).
NHRNSS	NUMBER OF HARNESS-BELT SYSTEMS TO BE SUPPLIED ON CARDS F.8.B-F.8.D. MAY BE ZERO OR BLANK. MAXIMUM VALUE = 5. NOTE: IN VERSION 12 (FOR WPAFB) THIS VARIABLE WAS SUPPLIED ON CARD F.8.A.
NWINDF	THE NUMBER OF WIND FORCE FUNCTIONS TO BE SUPPLIED ON CARDS E.6.A-E.6.N. MAY BE ZERO. NOTE: IN VERSION 12, THIS VARIABLE WAS SUPPLIED ON CARD E.5.
NJNTF	THE NUMBER OF JOINT RESTORING FORCE FUNCTIONS TO BE SUPPLIED ON CARDS E.7.A-E.7.N. MAY BE BLANK OR ZERO. NOTE: IN VERSION 12, THIS VARIABLE WAS SUPPLIED ON CARD E.5.

IF NPL IS NONZERO ON CARD D.1, NPL SETS OF CARDS D.2 ARE REQUIRED.

CARD D.2.A	FORMAT (I4, 4X, 5A4)
J	THE PLANE IDENTIFICATION NUMBER, MUST BE SUPPLIED AS CONSECUTIVE INTEGERS 1 TO NPL.
PLTTL(I,J),I=1,5	A 20 CHARACTER DESCRIPTION OF THE JTH PANEL.
CARDS D.2.B - D.2.D	FORMAT (3F12.0)
P1(I),I=1,3	THE X,Y AND Z COORDINATES OF POINT P1 IN VEHICLE (OR SEGMENT TO WHICH PLANE IS ATTACHED) REFERENCE (IN).
P2(I),I=1,3	THE X,Y AND Z COORDINATES OF POINT P2 IN VEHICLE (OR SEGMENT TO WHICH PLANE IS ATTACHED) REFERENCE (IN).
P3(I),I=1,3	THE X,Y AND Z COORDINATES OF POINT P3 IN VEHICLE (OR SEGMENT TO WHICH PLANE IS ATTACHED) REFERENCE (IN).

WHERE P1, P2 AND P3 ARE THREE OF THE CORNERS OF A PARALLELOGRAM SUCH THAT THE EDGE P1P2 IS LESS THAN 180 DEGREES CLOCKWISE (AS VIEWED FROM THE EXTERNAL SURFACE) FROM THE EDGE P1P3. NOTE: ANY PREVIOUS INPUT DECK IN WHICH THE VECTOR P2-P1 IS NOT PERPENDICULAR TO THE VECTOR P3-P1 WILL NOW PRODUCE DIFFERENT RESULTS.

IF NBLT IS NONZERO ON CARD D.1, NBLT SETS OF CARDS D.3 ARE REQUIRED.

CARD D.3.A		FORMAT (5A4)
	BLTTTL(I,J),I=1,5	A 20 CHARACTER DESCRIPTION OF THE JTH BELT.
CARD D.3.B		FORMAT (6F12.0)
	BELT(I,J),I=1,3	X,Y, AND Z COORDINATES, IN VEHICLE (OR SEGMENT TO WHICH BELT IS ANCHORED) REFERENCE, OF ANCHOR POINT A FOR THE JTH BELT (IN).
	BELT(I,J),I=4,6	X,Y, AND Z COORDINATES, IN VEHICLE (OR SEGMENT TO WHICH BELT IS ANCHORED) REFERENCE, OF ANCHOR POINT B FOR THE JTH BELT (IN).

NOTE: THE PROGRAM MUST PASS A PLANE THROUGH THE THREE POINTS, ANCHOR POINT A, ANCHOR POINT B, AND A FIXED POINT ON THE CONTACTED BODY SEGMENT. IF ANCHOR POINTS A AND B COINCIDE, THEY MUST BE SEPARATED SLIGHTLY FOR INPUT SUCH THAT THE DESIRED BELT PLANE WILL BE DEFINED.

CARD D.3.C		FORMAT (5F12.0)
	BELT(I,J),I=7,9	X, Y, AND Z COORDINATES, IN LOCAL BODY SEGMENT REFERENCE (BUT WITH RESPECT TO ELLIPSOID CENTER, NOT C.G.), OF THE FIXED CONTACT POINT ON THE BODY SEGMENT FOR THE JTH BELT (IN).
	BELT(10,J)	CURRENTLY NOT USED BY THE PROGRAM.
	BELT(11,J)	BELT SLACK (IN). THE SLACK, WHEN ADDED TO THE INITIAL GEOMETRIC LENGTH, RESULTS IN THE INITIAL BELT LENGTH. IF DESIRED, THE INITIAL BELT LENGTH MAY BE INPUTTED AS A NEGATIVE NUMBER AND THE PROGRAM WILL COMPUTE THE SLACK.

IF NBAG IS NONZERO ON CARD D.1, NBAG SETS OF CARDS D.4 ARE REQUIRED BY SUBROUTINE AIRBG1.

CARD D.4.A	FORMAT (5A4, I4)
BAGTTL(I,J),I=1,5	A 20 CHARACTER DESCRIPTION OF THE JTH AIR BAG.
NPANEL(J)	NUMBER OF VEHICLE CONTACT PANELS THAT ARE ALLOWED TO INTERACT WITH THE JTH AIR BAG (MAXIMUM = 4).
CARD D.4.B	FORMAT (6F12.0)
AB(I,J),I=1,3	THE X, Y AND Z SEMIAXES OF THE JTH AIR BAG WHEN FULLY INFLATED AND UNDEFORMED (IN).
BFA(I,J),I=1,3	THE X,Y AND Z COORDINATES OF THE CENTER OF THE AIR BAG CONTACT ELLIPSOID WITH RESPECT TO THE AIR BAG CENTER OF GRAVITY (IN).
CARD D.4.C	FORMAT (6F12.0)
YB,PB,RB	THE INITIAL ORIENTATION (YAW, PITCH, AND ROLL) OF THE JTH AIR BAG IN THE VEHICLE REFERENCE (DEG).
ZDEP(I,J),I=1,3	THE X, Y, AND Z COORDINATES OF THE DEPLOYMENT POINT OF THE JTH AIR BAG IN THE LOCAL REFERENCE OF THE 1ST PANEL ON CARD D.4.G (IN).
CARD D.4.D	FORMAT (6F12.0)
XBM(J)	WEIGHT OF AIR BAG MEMBRANE AND CONTENTS (LBS).
CYTD(J)	GAS SUPPLY ACTUATOR FIRING TIME AFTER THE START OF VEHICLE DECELERATION (SEC).
CYPA(J)	ATMOSPHERIC PRESSURE (PSIA).
CYSP(J)	INITIAL GAS SUPPLY PRESSURE (PSIG).
CYTO(J)	INITIAL GAS SUPPLY TEMPERATURE (DEG R).
CYVO(J)	GAS SUPPLY RESERVOIR VOLUME (IN**3).

CARD D.4.E	FORMAT (6F12.0)
CYCD(J)	SONIC THROAT DISCHARGE COEFFICIENT (DIMENSIONLESS).
CYK(J)	RATIO OF SPECIFIC HEATS OF SUPPLY GAS (DIMENSIONLESS).
CYR(J)	SPECIFIC GAS CONSTANT (IN/DEG R).
CYAT(J)	SONIC THROAT AREA (IN**2).
CYPV(J)	VENT PRESSURE OF THE EXHAUST ORIFICE (PSIG).
CYCDO(J)	EXHAUST ORIFICE DISCHARGE COEFFICIENT (DIMENSIONLESS).

CARD D.4.F	FORMAT (5F12.0)
CYAO(J)	EXHAUST ORIFICE AREA (IN**2).
SPRK(J)	SPRING CONSTANT OF A LINEAR SPRING USED TO SIMULATE ATTACHMENT OF THE BAG AT THE DEPLOYMENT POINT IN THE VEHICLE (LB/IN).
VSCS(J)	COEFFICIENT OF SLIDING FRICTION OF THE AIR BAG (DIMENSIONLESS)
CK(J)	PARAMETER USED TO STABILIZE AIR BAG NUMERICAL INTEGRATION (SEC**-1). SUGGESTED VALUE = 250.
CMASS(J)	MULTIPLIER TO INCREASE OR DECREASE THE MASS OF THE AIR BAG TO ARTIFICIALLY DAMPEN THE INTEGRATED AIR BAG MOTION.

NPANEL(J) SETS OF THE FOLLOWING TWO CARDS ARE REQUIRED TO DEFINE THE ELLIPSOIDS USED TO APPROXIMATE THE CONTACT PANELS FOR THE JTH AIR BAG. THE FIRST PANEL IS THE REACTION PANEL.

CARD D.4.G — FORMAT (6F12.0)

B(I,K,J),I=1,3	X, Y, AND Z SEMIAXES FOR THE KTH PANEL FOR THE JTH AIR BAG (IN).
BFB(I,K,J),I=1,3	THE LOCATION OF THE CENTER OF THE PANEL ELLIPSOID WITH RESPECT TO ITS CENTER OF GRAVITY (IN).

CARD D.4.H — FORMAT (6F12.0)

ZR(I,K,J),I=1,3	X, Y, AND Z COORDINATES IN VEHICLE REFERENCE OF THE CENTER OF GRAVITY OF THE KTH PANEL OF THE JTH AIR BAG (IN).
YP,PP,RP	ANGULAR ORIENTATION, YAW,PITCH AND ROLL (DEG.), OF THE KTH PANEL WITH RESPECT TO THE VEHICLE.

IF NELP IS NONZERO ON CARD D.1, NELP D.5 CARDS ARE REQUIRED I
BY SUBROUTINE BINPUT. I

NOTE: NELP IS THE NUMBER OF CONTACT ELLIPSOIDS TO BE SUPPLIED HERE, NOT THE NUMBER OF CONTACT ELLIPSOIDS IN THE PROGRAM. THE FIRST NSEG ELLIPSOIDS WERE SUPPLIED ON CARDS B.2.A - B.2.I WITH NO ANGULAR ROTATIONS. THEY MAY BE REPLACED HERE IF DESIRED.

CARDS D.5.A - D.5.J (NELP CARDS)	FORMAT (I6, 9F6.0)
M	CONTACT ELLIPSOID NUMBER. MAX = 40. IF M < NSEG + 1, DATA WILL REPLACE INPUT SUPPLIED ON CARDS B.2A - B.2.I. OTHERWISE, M MUST BE GREATER THAN NSEG+NBAG+1. I S S
P1(I),I=1,3	THE X, Y, AND Z SEMIAXES OF THE CONTACT ELLIPSOID (IN).
P2(I),I=1,3	THE X, Y, AND Z COORDINATES OF THE ELLIPSOID OFFSET FROM THE SEGMENT CENTER OF GRAVITY.
P3(I),I=1,3	THE YAW, PITCH AND ROLL (DEGREES) OF THE CONTACT ELLIPSOID FROM THE PRINCIPAL AXIS OF THE SEGMENT.

IF NQ IS NONZERO ON CARD D.1, NQ D.6 CARDS ARE REQUIRED.

CARDS D.6.A - D.6.J (NQ CARDS) FORMAT (3I6, 6F6.0)

KQTYPE(J)	TYPE NO. OF THE JTH CONSTRAINT 1: POINT SPECIFIED BY RK1 ON SEGMENT KQ1 WILL BE CONSTRAINED TO BE THE SAME AS THE POINT SPECIFIED BY RK2 ON SEGMENT KQ2. 2: POINT SPECIFIED BY RK1 ON SEGMENT KQ1 WILL BE CONSTRAINED TO REMAIN AT AN EQUAL DISTANCE (D > 0) FROM THE POINT SPECIFIED BY RK2 ON SEGMENT KQ2. 5: TENSION ELEMENT CONSTRAINT CONNECTING POINT RK1 ON SEGMENT KQ1 TO POINT RK2 ON SEGMENT RK2 (REQUIRES TWO CARDS WITH KQTYPE, KQ1 AND KQ2 THE SAME ON BOTH).
KQ1(J)	SEGMENT IDENTIFICATION NUMBER OF THE 1ST SPECIFIED POINT.
KQ2(J)	SEGMENT IDENTIFICATION NUMBER OF THE 2ND SPECIFIED POINT.
RK1(I,J),I=1,3	COORDINATES OF SPECIFIED POINT ON SEGMENT KQ1 (IN). IF KQTYPE = 5, THE SECOND CARD WILL CONTAIN THE EFFECTIVE MASSES MA, MB AND MAB (LB.SEC**2/IN) IN PLACE OF RK1.
RK2(I,J),I=1,3	COORDINATES OF SPECIFIED POINT ON SEGMENT KQ2 (IN). IF KQTYPE = 5, THE SECOND CARD WILL CONTAIN THE SPRING CONSTANT K (LB/IN), THE VISCOUS DAMPING CONSTANT D (LB SEC/IN) AND THE REFERENCE LENGTH L (IN) IN PLACE OF RK2. NOTE: IF KQTYPE = 1 AND KQ2 IS THE NUMBER FOR THE VEHICLE, THEN SUBROUTINE EQUILB WILL MODIFY THESE VALUES OF RK2 SUCH THAT THEY WILL BE EQUIVALENT TO RK1 IN INERTIAL REFERENCE FOR TIME ZERO. (SEE DESCRIPTION UNDER CARDS G.6.)

CARD D.7 IS ALWAYS REQUIRED. SUPPLY BLANK CARD FOR NORMAL 3D MOTION.

CARD D.7	FORMAT (18I4)	IF NSEG>18, USE 2 CARDS.

NSYM(J),J=1,NSEG — CONTROLS SYMMETRY OPTION OF BODY SEGMENTS AS FOLLOWS :

NSYM(J) = 0 : NORMAL THREE DIMENSIONAL MOTION FOR BODY SEGMENT J.

NSYM(J) = J : MOTION OF BODY SEGMENT J WILL BE RESTRICTED TO THE X-Z PLANE WITH NO LATERAL MOTION, HENCE IT WILL BE TWO DIMENSIONAL.

NSYM(J) = K : BODY SEGMENTS J AND K ARE TO REMAIN SYMMETRICAL WITH NO LATERAL MOTION. THE MOTION OF EACH WILL BE REPLACED WITH THEIR AVERAGE AND RESTRICTED TO THE LOCAL X-Z PLANE. NSYM(K) MUST EQUAL J.

NSYM(J) = -K : BODY SEGMENTS J AND K ARE TO REMAIN MIRROR SYMMETRICAL WITH RESPECT TO THE X-Z PLANE. EQUAL BUT OPPOSITE LATERAL MOTION IS PERMITTED. NSYM(K) MUST EQUAL -J.

NOTE : IN THE ABOVE SYMMETRY OPTIONS, THE USER MUST TAKE EXTREME CARE THAT ALL INPUT WILL ALLOW THE SYMMETRY TO EXIST.

```
IF NSD IS NONZERO ON CARD D.1, NSD D.8 CARDS ARE REQUIRED.

CARDS D.8.A - D.8.J       FORMAT (2I3, 11F6.0)
      (NSD CARDS)

    MSDM(J)               SEGMENT IDENTIFICATION NUMBERS (M AND N)
    MSDN(J)               TO WHICH THE JTH SPRING DAMPER IS ATTACHED.

    APSDM(I,J),I=1,3      COORDINATES OF ATTACHMENT POINTS IN LOCAL
    APSDN(I,J),I=1,3      SEGMENT REFERENCE ON SEGMENTS M AND N FOR
                          THE JTH SPRING DAMPER (IN.)

    ASD(I,J),I=1,5        COEFFICIENTS OF QUADRATIC FUNCTIONS TO
 I=1 : D0 (IN)            COMPUTE THE SPRING FORCE (FS) AND THE
 I=2 : A1 (LB/IN)         VISCOUS FORCE (FD) FOR THE JTH SPRING
 I=3 : A2 (LB/IN**2)      DAMPER USING THE RELATIONSHIPS
 I=4 : B1 (LB SEC/IN)
 I=5 : B2 (LB SEC**2/IN**2)
                                 FS= (D-D0)*(|A1| + A2*|D-D0|)
                                 FD=      DV*( B1  + B2*|DV|)

                          WHERE D AND DV ARE THE DISTANCE AND ITS TIME
                          DERIVATIVE BETWEEN THE POINTS APSDM AND APSDN.
                          IF A1 < 0.  AND  (D-D0) < 0.,
                          PROGRAM WILL SET FS= 0., I.E. THIS WILL ACT AS A
                          TENSION ELEMENT.
```

E. SUBROUTINE CINPUT (FUNCTIONS INPUT)

THESE FUNCTIONS ARE REFERRED TO BY NUMBER IN THE NF ARRAYS REQUIRED ON CARDS F.1.B, F.2.B, F.3.B AND F.4.B. THEY ARE USED TO DEFINE THE FORCE DEFLECTION, INERTIAL SPIKE, R (ENERGY ABSORPTION) FACTOR, G (DEFLECTION) FACTOR AND FRICTION COEFFICENT FUNCTIONS.

EACH FUNCTION MAY BE SUBDIVIDED, IF DESIRED, INTO TWO SEPARATE PARTS, F1 AND F2, WHERE

F1(D) IS DEFINED FOR 0 .LE. D0 .LE. D .LE. |D1|

F2(D) IS DEFINED FOR |D1| .LE. D .LE. |D2|.

IN ADDITION, EACH PART OF A FUNCTION MAY BE DEFINED BY EITHER OF THREE FUNCTIONAL FORMS: CONSTANT VALUE, TABULAR DATA OR A FIFTH DEGREE POLYNOMIAL. THE EXISTENCE AND FORM OF EACH PART IS DETERMINED BY THE SUPPLIED VALUES OF D0, D1 AND D2 AS FOLLOWS:

F1	F2	D0	D1	D2
CONSTANT	-	0	0	F1 = D2
TABULAR	-	D0 .GE. 0	D1 .LT. 0	0
POLYNOMIAL	-	D0 .GE. 0	D1 .GT. 0	0
TABULAR	POLYNOMIAL	D0 .GE. 0	D1 .LT. 0	D2 .GT. 0
POLYNOMIAL	TABULAR	D0 .GE. 0	D1 .GT. 0	D2 .LT. 0
POLYNOMIAL	POLYNOMIAL	D0 .GE. 0	D1 .GT. 0	D2 .GT. 0

THE CONSTANT FORM IS APPLICABLE TO F1 ONLY BECAUSE THE ROUTINES ASSUME

IF D .GT. |D2| THEN F(D) = F(|D2|) FOR D2 .NE. 0 OR

IF D .GT. |D1| THEN F(D) = F(|D1|) FOR D2 = 0.

THE CASE OF BOTH F1 AND F2 BEING TABULAR IS UNNECESSARY.

A MAXIMUM OF 50 FUNCTIONS MAY BE SUPPLIED TO THE PROGRAM. THESE FUNCTIONS MAY BE OF THE TYPES DESCRIBED ON EITHER CARDS E.1-E.4, CARDS E.6 OR CARDS E.7.

CARD E.1	FORMAT (I4, 4X, 5A4)
I	THE FUNCTION IDENTIFYING NUMBER. THESE NUMBERS NEED NOT BE SUPPLIED IN NUMERIC ORDER. IF THE SAME NUMBER IS USED MORE THAN ONCE, A WARNING WILL BE PRINTED AND THE LAST ONE SUPPLIED WILL BE USED. THE END OF THE FUNCTION INPUT IS INDICATED BY SUPPLYING A SINGLE CARD WITH I > 50.
KTITLE	A 20 CHARACTER ALPHANUMERIC TITLE DESCRIBING THE FUNCTION.

CARD E.2 FORMAT (5F12.0)

D0 THE LOWER ABSCISSA VALUE OF THE FIRST PART (F1) OF THE FUNCTION. UNITS ARE DEPENDENT ON USAGE OF THE FUNCTION, I.E. IN. FOR DEFLECTION, IN./IN. FOR STRESS-STRAIN, IN/SEC FOR RATE DEPENDENT FUNCTIONS. NORMALLY A VALUE OF ZERO IS USED FOR FORCE DEFLECTION FUNCTIONS. A NEGATIVE VALUE MAY BE SUPPLIED FOR RATE DEPENDENT FUNCTIONS.

D1 THE MAGNITUDE OF D1 IS THE UPPER ABSCISSA VALUE OF F1 AND THE LOWER ABSCISSA VALUE OF F2, IF ANY. D1 < 0 INDICATES F1 IS TABULAR, D1 > 0 INDICATES F1 IS A POLYNOMIAL, AND D1 = 0 INDICATES F1 = D2, A CONSTANT.

D2 IF D1 = 0, D2 IS THE CONSTANT VALUE OF F1. OTHERWISE, THE MAGNITUDE OF D2 IS THE UPPER ABSCISSA VALUE OF F2. IF D2 = 0, F2 IS NOT DEFINED; IF D2 IS NEGATIVE, F2 IS TABULAR; AND IF D2 IS POSITIVE, F2 IS A POLYNOMIAL.

D3 IF THE FUNCTION IS TO BE USED FOR AN INERTIAL SPIKE, D3 REPRESENTS THE ABSCISSA VALUE FOR WHICH THE INERTIAL SPIKE IS TO BE IGNORED IF UNLOADING OCCURS AFTER DEFLECTION EXCEEDS D3. IF THE FUNCTION IS TO BE USED FOR A COEFFICIENT OF FRICTION, D3 = (1+U)/2 WHERE U IS THE COEFFICIENT OF RESTITUTION FOR THE IMPULSE OPTION (0<D3<1 OR -1<U<+1). A VALUE OF D3 = 0 MEANS THAT THE IMPULSE OPTION WILL NOT BE USED FOR THOSE CONTACTS USING THIS FUNCTION. WHEN THE GLOBALGRAPHIC OPTION IS USED, A FRICTION FUNCTION IS DEFINED AND THE VALUE OF D3 IS USED TO SPECIFY THE IMPULSE. (SEE CARD B.5.)

D4 IF THE FUNCTION IS TO BE USED AS A FORCE DEFLECTION FUNCTION BY SUBROUTINE PLELP, D4=RHO, THE SCALAR THAT DETERMINES THE POINT OF FORCE APPLICATION. SUPPLY ZERO FOR POINT OF MAXIMUM PENETRATION, ONE FOR CENTER OF INTERSECTION ELLIPSE. IF USED AS THE FRICTION FUNCTION FOR A ROLL-SLIDE CONSTRAINT. D4 IS THE COEFFICIENT OF STATIC FRICTION TO BE USED FOR THE ROLL CONSTRAINT.

THE DEFINITIONS OF F1 AND F2, IF THEY EXIST, ARE NOW SUPPLIED ON CARD E.3 FOR THE FIFTH DEGREE POLYNOMIAL DEFINITION, OR ON CARDS E.4 FOR THE TABULAR DEFINITION.

CARD E.3	FORMAT (6F12.0)
A0,A1,A2,A3,A4,A5	COEFFICIENTS OF FIFTH-DEGREE POLYNOMIAL F = A0 + A1*X + A2*X**2 + A3*X**3 + A4*X**4 + A5*X**5 (UNITS ARE DEPENDENT ON USE OF FUNCTION.)
CARD E.4.A	FORMAT (I6)
NPI	THE NUMBER OF DATA POINTS TO BE SUPPLIED TO IDENTIFY THE FUNCTION IF IT IS DEFINED IN TABULAR FORM.
CARDS E.4.B - E.4.N	FORMAT (6F12.0)
(X(I),Y(I),I=1,NPI)	THE ABSCISSA AND ORDINATE VALUES OF THE DATA POINTS USED TO DEFINE THE TABULAR FORM OF THE FUNCTION. THE PROGRAM WILL LINEARLY INTERPOLATE TO DETERMINE INTERMEDIATE VALUES. SUPPLY 3 POINTS PER CARD; USE AS MANY CARDS AS REQUIRED. (UNITS ARE DEPENDENT ON USE OF FUNCTION.)

NOTE: ALWAYS SUPPLY A CARD E.1 WITH A FUNCTION NUMBER > 50 AFTER ALL FUNCTIONS ARE DEFINED TO SIGNAL THE END OF FUNCTION INPUTS.

```
SUBROUTINE KINPUT (WIND FORCE AND JOINT RESTORING FORCE FUNCTIONS)

NOTE: CARD E.5, PREVIOUSLY REQUIRED FOR VERSION 12 (WPAFB CONTRACT
NO. F33615-75-C-5002 AS DOCUMENTED IN REPORT NO. AMRL-TR-75-14) IS
NO LONGER REQUIRED. THE VARIABLES NWINDF AND NJNTF ARE NOW SUPPLIED
ON CARD D.1.

IF NWINDF=0 ON CARD D.1, CARDS E.6 ARE NOT REQUIRED. OTHERWISE,
NWINDF SETS OF CARDS E.6.A - E.6.N ARE REQUIRED.

CARD E.6.A                  FORMAT (I4, 4X, 5A4)

    I,KTITLE                SAME AS CARD E.1 EXCEPT THAT EACH FUNCTION
                            NUMBER (I) MUST BE LESS THAN 51 AND MUST BE
                            DISTINCT FROM THOSE SUPPLIED ON CARDS E.1.

CARD E.6.B                  FORMAT (5F12.0)

    D0,D1,D2,D3,D4          CURRENTLY NOT USED BY PROGRAM.

CARD E.6.C                  FORMAT (I6)

    NTMPTS                  THE NUMBER OF TIME POINTS OR CARDS REQUIRED
                            TO DEFINE THIS FUNCTION ON CARDS E.6.D-E.6.N.

CARDS E.6.D - E.6.N         FORMAT (4F12.0)
   (NTMPTS CARDS)

    T                       TIME (SEC.) SINCE INITIAL PENETRATION OF
                            BOUNDARY PLANE. VALUES SHOULD BE IN ASCENDING
                            ORDER WITH FIRST VALUE EQUAL TO ZERO.

    FX,FY,FZ                THE X,Y AND Z COMPONENTS OF FORCE PER UNIT
                            AREA (LBS./IN.**2) IN INERTIAL REFERENCE
                            DUE TO THE WIND BLAST FORCE AT TIME T. THE
                            PROGRAM WILL USE LINEAR INTERPOLATION ON T.
                            IF LAST VALUE OF T IS EXCEEDED, THE LAST
                            VALUES OF FX,FY AND FZ WILL BE USED.
```

IF NJNTF=0 ON CARD D.1, CARDS E.7 ARE NOT REQUIRED. OTHERWISE, NJNTF (FROM CARD D.1) SETS OF CARDS E.7.A - E.7.N ARE REQUIRED.

CARD E.7.A	FORMAT (I4, 4X, 5A4)
I,KTITLE	SAME AS CARD E.1 EXCEPT THAT EACH FUNCTION NUMBER (I) MUST BE LESS THAN 51 AND MUST BE DISTINCT FROM THOSE SUPPLIED ON CARDS E.1 OR CARDS E.6.A.
CARD E.7.B	FORMAT (5F12.0)
D0,D1,D2,D3,D4	CURRENTLY NOT USED BY PROGRAM.
CARD E.7.C	FORMAT (2I6)
NTHETA	MAGNITUDE INDICATES THE NUMBER OF COLUMNS IN THE TWO DIMENSIONAL INPUT DATA MATRIX TO BE SUPPLIED ON CARDS E.7.D-E.7.N. THE MINIMUM VALUE IS 2. IF POSITIVE, THE NTHETA ENTRIES IN EACH ROW WILL BE TABULAR DATA FOR EQUALLY SPACED VALUES OF THE JOINT FLEXURE ANGLE (THETA) BETWEEN 0 AND 180 DEGREES. IF NEGATIVE, THE ENTRIES WILL REPRESENT THE COEFFICIENTS OF A (-NTHETA-1) ORDER POLYNOMIAL IN (THETA-THETA0)
NPHI	NUMBER OF ROWS OF MATRIX OF DATA TO BE SUPPLIED ON CARDS E.7.D-E.7.N. EACH ROW REPRESENTS EQUALLY SPACED VALUES OF THE JOINT AZIMUTH ANGLE (PHI) BETWEEN -180 AND +180 DEGREES, BUT DOES NOT INCLUDE THE LAST ROW SINCE THE PROGRAM ASSUMES DATA FOR PHI(NPHI+1)=180 ARE THE SAME AS FOR PHI(1)=-180. MINIMUM = 1.
CARDS E.7.D - E.7.N	FORMAT (5F12.0)

(NPHI SETS OF CARDS. USE EXTRA CARDS PER SET IF |NTHETA| > 5.)

THETA0	THE VALUE OF THE "DEAD BAND" ZONE FOR THIS VALUE OF PHI (DEGREES). IF THE FLEXURE ANGLE (THETA) IS LESS THAN THETA0, THE JOINT RESTORING FORCE WILL BE ZERO.
F(J),J=2,NTHETA	FOR NTHETA POSITIVE, TABULAR VALUES OF THE JOINT RESTORING FORCE FOR FLEXURE ANGLES THETA(J) = (J-1)*180/(NTHETA-1) DEGREES VALUES OF ZERO SHOULD BE SUPPLIED FOR THETA < THETA0. FOR NTHETA NEGATIVE, THE COEFFICIENTS OF A POLYNOMIAL IN (THETA-THETA0) OF ORDER ONE LESS THAN THE MAGNITUDE OF NTHETA. F(J) IS THE COEFFICIENT OF (THETA-THETA0)**(J-1) WHERE (THETA-THETA0) IS EXPRESSED IN RADIANS. F(1) IS ASSUMED TO BE ZERO.

F SUBROUTINE FINPUT (ALLOWED CONTACTS) I

IF NPL IS NONZERO ON CARD D.1, CARDS F.1 ARE REQUIRED. I

CARD F.1.A	FORMAT (18I4) IF NPL>18, USE 2 CARDS.
MNPL(J),J=1,NPL	FOR PLANE J, THE NUMBER OF SEGMENTS FOR WHICH SEGMENT-PLANE CONTACT IS ALLOWED. NPL IS THE NUMBER OF PLANES FROM CARD D.1. THE VALUE OF ANY MNPL FOR PLANE J MAY BE ZERO AND THE MAXIMUM VALUE IS 5. HOWEVER IF IT IS REQUIRED TO HAVE MORE THAN 5 SEGMENTS CONTACT THE SAME PLANE, SET UP TWO OR MORE IDENTICAL PLANES AND PERMIT A MAXIMUM OF 5 SEGMENTS TO CONTACT EACH PLANE.

FOR EACH PLANE J, MNPL(J) CARDS OF THE FOLLOWING MUST BE SUPPLIED.

CARDS F.1.B - F.1.N	FORMAT (9I4)
NJ	THE PLANE NUMBER FOR WHICH CONTACT IS ALLOWED. NJ MUST CORRESPOND TO J ABOVE. THERE MUST BE MNPL(J) CARDS WITH THIS SAME NJ. IF MNPL(J) = 0, NO NJ = J SHOULD BE PRESENT.
NS(1)	THE SEGMENT NUMBER TO WHICH PLANE J IS ATTACHED. IF VEHICLE, SUPPLY NSEG+1, IF GROUND, SUPPLY NSEG+NBAG+2. S
NS(2)	THE SEGMENT NUMBER (DETERMINED BY THE CARD NUMBER I UNDER CARD B.2.A) FOR WHICH CONTACT WITH THE NJTH PLANE IS ALLOWED.
NS(3)	THE NUMBER OF THE CONTACT ELLIPSOID ASSOCIATED WITH THE SEGMENT NS(2).

```
NF(1)                   THE FUNCTION NO. FROM CARD E.1 TO DEFINE THE
                        FORCE DEFLECTION FUNCTION FOR THIS CONTACT.
                        IF NF(1)=0, A ROLL-SLIDE CONSTRAINT WILL BE
                        EXERCISED BY THE PROGRAM FOR THIS CONTACT
                        WHICH DOES NOT REQUIRE NF(2),NF(3) OR NF(4)
                        BUT DOES REQUIRE A FRICTION COEFFICIENT
                        FUNCTION TO BE DEFINED BY NF(5). ALSO, THE
                        INITIAL POSITIONS ON CARDS G.2 MUST BE SUCH
                        THAT THERE IS NO CONTACT AT TIME = 0.

NF(2)                   THE FUNCTION NO. FROM CARD F.1 TO DEFINE THE
                        INERTIAL SPIKE FUNCTION FOR THIS CONTACT.
                        IF ZERO OR NEGATIVE, NO INERTIAL SPIKE EXISTS.   *
                        IF NEGATIVE, THE MAGNITUDE SPECIFIES THE         *
                        FUNCTION NO. FOR F2 OF THE RATE DEPENDENT        *
                        FUNCTIONS DESCRIBED BELOW.                       *

NF(3)                   THE FUNCTION NO. FROM CARD E.1 TO DEFINE THE
                        R (ENERGY ABSORPTION) FACTOR FUNCTION. A VALUE
                        OF R=1 INDICATES THAT ALL ENERGY IS RECOVERED
                        (NO LOSS) AND R=0 THAT NO ENERGY IS RECOVERED.
                        IF ZERO OR NEGATIVE, R=1 IS ASSUMED (DEFAULT).   *
                        IF NEGATIVE, THE MAGNITUDE SPECIFIES THE         *
                        FUNCTION NO. FOR F3 OF THE RATE DEPENDENT        *
                        FUNCTIONS DESCRIBED BELOW.                       *

NF(4)                   THE FUNCTION NO. FROM CARD E.1 TO DEFINE THE
                        G (PERMANENT DEFLECTION) FACTOR FUNCTION.
                        IF ZERO OR NEGATIVE, G=0 IS ASSUMED (DEFAULT).   *
                        IF NEGATIVE, THE MAGNITUDE SPECIFIES THE         *
                        FUNCTION NO. FOR F4 OF THE RATE DEPENDENT        *
                        FUNCTIONS DESCRIBED BELOW.                       *

NF(5)                   THE FUNCTION NO. FROM CARD E.1 TO DEFINE THE
                        FRICTION COEFFICIENT FUNCTION. IF FOR A ROLL-
                        SLIDE CONSTRAINT (NF(1)=0), THE VALUE OF D3
                        ON CARD E.2 FOR THIS FUNCTION SHOULD BE 0.5.

NOTE: RATE DEPENDENT FUNCTIONS CAN BE USED INSTEAD OF THE INERTIAL       *
SPIKE, R AND G FACTORS BY DEFINING NF(2), NF(3) AND NF(4) ALL ZERO       *
OR NEGATIVE. THE TOTAL FORCE DEFLECTION FUNCTION IS COMPUTED BY          *
                                                                         *
            F(D,D') = F1(D) + F2(D)*F3(D') + F4(D')                      *
                                                                         *
WHERE D AND D' ARE THE DEFLECTION AND RATE OF DEFLECTION; AND F1,F2,     *
F3 AND F4 ARE FUNCTIONS SPECIFIED BY NF(1),NF(2),NF(3) AND NF(4).        *
IF NF(2),NF(3) OR NF(4) IS ZERO, THE CORRESPONDING FUNCTION IS ZERO.     *
IF D<0, THE RATE DEPENDENT FUNCTIONS ARE NOT COMPUTED AND F(D,D')=0.     *
THE FUNCTIONS SHOULD BE DEFINED SUCH THAT F1(D), F2(D), D'*F3(D')        *
AND D'*F4(D') ARE ALL GREATER THAN OR EQUAL TO ZERO. HENCE, F(D,D')      *
MAY BE NEGATIVE IF D' IS NEGATIVE.                                       *
```

IF NBLT IS NONZERO ON CARD D.1, CARDS F.2 ARE REQUIRED.

CARD F.2.A FORMAT (8I4)

MNBLT(J),J=1,NBLT	FOR BELT J, THE NUMBER OF SEGMENTS FOR WHICH SEGMENT-BELT INTERACTION IS ALLOWED. NBLT IS THE NUMBER OF BELTS FROM CARD D.1. EACH MNBLT MAY HAVE A VALUE OF 0 OR 1 ONLY.

FOR EACH BELT J, MNBLT(J) CARDS OF THE FOLLOWING MUST BE SUPPLIED.

CARDS F.2.B - F.2.N FORMAT (9I4)

NJ	THE BELT NUMBER TO BE CONTACTED, MUST CORRESPOND TO J ABOVE. THERE MUST BE MNBLT(J) CARDS WITH THE SAME NJ. IF MNBLT(J) = 0, NO NJ = J SHOULD BE PRESENT.
NS(1)	THE SEGMENT NUMBER TO WHICH BELT NJ IS ATTACHED. IF VEHICLE, SUPPLY NSEG+1. IF GROUND, SUPPLY NSEG+NBAG+2.
NS(2)	THE SEGMENT NUMBER (DETERMINED BY THE CARD NUMBER I UNDER CARD B.2.A) FOR WHICH INTERACTION WITH THE NJTH BELT IS ALLOWED.
NS(3)	THE NUMBER OF THE CONTACT ELLIPSOID ASSOCIATED WITH THE SEGMENT NS(2).
NF(1)	THE FUNCTION NUMBER FROM CARD E.1 TO DEFINE THE FORCE-DEFLECTION FUNCTION FOR THIS CONTACT. THE ABSCISSA FOR THIS FUNCTION SHOULD BE STRAIN (IN/IN).
NF(I),I=2,4	SAME DEFINITION AS ON CARD F.1.B ABOVE.
NF(5)	IF NON-ZERO, FULL BELT FRICTION IS ASSUMED, I.E., FORCES ARE COMPUTED FOR EACH HALF OF THE BELT SEPARATELY. IF ZERO, ZERO BELT FRICTION IS ASSUMED, I.E., BELT TENSION IS IS THE SAME AT BOTH BELT ANCHOR POINTS.

NOTE: THE USE OF RATE DEPENDENT FUNCTIONS AS DEFINED UNDER CARDS F.1.B ARE NOT CURRENTLY OPERATIONAL FOR BELT-SEGMENTS CONTACTS.

CARD F.3.A IS ALWAYS REQUIRED. MAY BE BLANK TO SPECIFY THAT NO SEGMENT-SEGMENTS ARE TO BE COMPUTED BY THE PROGRAM.

CARD F.3.A	FORMAT (18I4) IF NSEG>18, USE TWO CARDS.
MNSEG(J),J=1,NSEG	FOR SEGMENT J, THE NUMBER OF SEGMENTS FOR WHICH SEGMENT-SEGMENT CONTACT IS ALLOWED. NSEG IS THE NUMBER OF SEGMENTS FROM CARD B.1. EACH SEGMENT CONTACT, A VERSUS B, MAY BE INPUTTED EITHER WAY EXCEPT WHERE AN INTERIOR CONTACT IS DESIRED (SEE NS(3)). ANY OR ALL VALUES OF MNSEG MAY BE ZERO. THE MAXIMUM VALUE FOR EACH MNSEG IS 5.

FOR EACH SEGMENT J, MNSEG(J) CARDS OF THE FOLLOWING MUST BE SUPPLIED.

CARDS F.3.B - F.3.N	FORMAT (9I4)
NJ	THE SEGMENT NUMBER TO BE CONTACTED, MUST CORRESPOND TO J ABOVE. THERE MUST BE MNSEG(J) CARDS WITH THIS SAME NJ. IF MNSEG(J) = 0, NO NJ = J SHOULD BE PRESENT.
NS(1)	THE NUMBER OF THE CONTACT ELLIPSOID ASSOCIATED WITH SEGMENT NJ.
NS(2)	THE SEGMENT NUMBER (DETERMINED BY THE CARD NUMBER I UNDER CARD B.2.A) FOR WHICH CONTACT WITH THE NJTH SEGMENT IS ALLOWED.
NS(3)	THE NUMBER OF THE CONTACT ELLIPSOID ASSOCIATED WITH THE SEGMENT NS(2). IF NEGATIVE, AN INTERIOR CONTACT WILL BE ASSUMED WITH ELLIPSOID NS(1) INSIDE NS(3).
NF(I),I=1,5	SAME DEFINITIONS AS ON CARD F.1.B ABOVE.

NOTE: THE USE OF RATE DEPENDENT FUNCTIONS AS DEFINED UNDER CARDS F.1.B ARE PERMISSABLE FOR SEGMENT-SEGMENT CONTACTS.

IF NJNT IS NONZERO ON CARD B.1, CARD F.4.A IS REQUIRED.
SUPPLY IGLOB=1 FOR GLOBALGRAPHIC OPTION, OTHERWISE SUPPLY 0 OR BLANK

CARD F.4.A — FORMAT (18I4) IF NJNT>18, USE TWO CARDS.

IGLOB(J),J=1,NJNT — FOR EACH JOINT J, SUPPLY 1 FOR IGLOB(J) IF IPIN(J) IS +3 OR -3 ON CARDS B.3.A - B.3.J; OTHERWISE SUPPLY ZERO OR BLANK. ONE CARD F.4.J MUST BE SUPPLIED BELOW FOR EACH J FOR WHICH IGLOB(J) =1.

CARDS F.4.B - F.4.J — FORMAT (9I4)

NJ — THE IDENTIFICATION NUMBER FOR A GLOBALGRAPHIC JOINT, MUST CORRESPOND TO J ABOVE AND CARDS MUST BE SUPPLIED IN ASCENDING ORDER ON NJ.

NS(I),I=1,3 — CURRENTLY NOT USED BY PROGRAM.

NF(1) — THE FUNCTION NUMBER FROM CARD E.1 TO DEFINE THE TORQUE-DEFLECTION FOR THIS GLOBALGRAPHIC JOINT. THE ORDINATE FOR THIS FUNCTION SHOULD BE TORQUE (IN. LB.) AND THE ABSCISSA IS THE ANGULAR DEFLECTION (RADIANS) INTO THE STOP.

NF(2) — THE FUNCTION NUMBER FROM CARD E.1 TO DEFINE THE HERRON FORMULAS FOR T (JOINT STOP ANGLE IN RADIANS) AND ITS DERIVATIVE TP WITH RESPECT TO PHI BOTH AS FUNCTIONS OF PHI (THE JOINT ANGLE FROM THE REFERENCE AXIS IN RADIANS). NORMALLY THEY WILL BE COMPUTED BY

```
 T = P1 + SP*P2
TP = P1' + CP*P2 + SP*P2'
```

WHERE P1,P2 ARE THE 5TH DEGREE POLYNOMIAL EVALUATIONS OF COS(PHI) USING THE TWO POLYNOMIALS F1 AND F2 OBTAINED BY SETTING BOTH D1,D2 > 0 ON CARD E.2;

P1',P2' ARE THEIR DERIVATIVES WITH RESPECT TO PHI;

AND CP,SP ARE COS(PHI) AND SIN(PHI).

IF D1,D2 ARE NOT BOTH POSITIVE, T AND TP WILL BE EVALUATED AS FUNCTIONS OF PHI IN RADIANS (0 < PHI < 2*PI) AS SPECIFIED ON CARDS E.1 - E.4 FOR FUNCTION NF(2).

NF(I),I=3,5 — SAME DEFINITIONS AS ON CARD F.1.B ABOVE EXCEPT THAT THE USE OF RATE DEPENDENT FUNCTIONS IS NOT PERMITTED.

IF NJNT>0 (CARD B.1) AND NJNTF>0 (CARD D.1), CARD F.5.A IS REQUIRED. IF NJNT>0 AND NJNTF=0, THE PROGRAM WILL SET THE JOINTF ARRAY TO ZERO AND CARD F.5.A IS NOT REQUIRED (NOTE: FOR VERSION 12 A BLANK CARD WAS REQUIRED).

CARD F.5.A	FORMAT (18I4) USE TWO CARDS IF NJNT > 18.
JOINTF(J),J=1,NJNT	FOR EACH JOINT (J), THE FUNCTION IDENTIFICATION NUMBER AS SUPPLIED ON CARDS E.7.A TO BE USED BY SUBROUTINE VISPR TO COMPUTE THE JOINT RESTORING FORCE BY FUNCTION FNTERP. IF ZERO, THE VALUES OF SPRING(1,3*J-2) AS SUPPLIED ON CARDS B.4.A WILL BE USED USING FUNCTION EJOINT.

IF NBAG # 0, NBAG CARDS OF THE FOLLOWING MUST BE SUPPLIED. SINCE THE AIR BAG ROUTINES DO NOT USE THE FORCE-DEFLECTION FUNCTIONS, THIS INPUT HAS DIFFERENT FORMATS THAN THE ABOVE ALLOWED CONTACTS.

CARDS F.6.A - F.6.N	FORMAT (2I4, 20I2)
K	THE AIR BAG NUMBER CORRESPONDING TO THE INDEX J UNDER CARDS D.4 ABOVE. K MUST BE IN NUMERIC ORDER K = 1 TO NBAG, WHERE NBAG IS THE NUMBER OF AIR BAGS DEFINED ON CARD D.1.
NK	THE NUMBER OF SEGMENTS ALLOWED TO CONTACT THE KTH AIR BAG. THE MAXIMUM VALUE IS 10. IF NK = 0, THE REMAINDER OF THE CARD IS BLANK.
MBAG(2,I,K), MBAG(3,I,K),I=1,NK	THE SEGMENT NUMBERS (DETERMINED BY THE CARD NUMBER I UNDER CARD B.2.A) EACH FOLLOWED BY THE NUMBER OF THE ASSOCIATED CONTACT ELLIPSOID FOR WHICH CONTACT FORCES WITH THE KTH AIR BAG WILL BE COMPUTED.

IF NWINDF=0 ON CARD D.1, CARDS F.7 ARE NOT REQUIRED AND THE PROGRAM WILL SET THE MWSEG ARRAY TO ZEROS (NOTE: FOR VERSION 12 A BLANK CARD F.7.A WAS PREVIOUSLY REQUIRED). OTHERWISE, CARDS F.7 ARE REQUIRED.

CARD F.7.A	FORMAT (18I4) USE TWO CARDS IF NSEG > 18.
MWSEG(1,J),J=1,NSEG	FOR EACH SEGMENT J, SUPPLY ZERO IF NO WIND FORCE CALCULATIONS ARE TO BE PERFORMED. OTHERWISE, SUPPLY A VALUE OF ONE TO INDICATE WIND FORCES ARE TO BE PERFORMED.

SUPPLY CARD F.7.B FOR EACH SEGMENT (J) WHERE MWSEG(1,J) = 1.

CARD F.7.B	FORMAT (5I4)
JJ	THE SEGMENT IDENTIFICATION NUMBER FROM CARDS B.2.A FOR WHICH WIND FORCE CALCULATIONS ARE TO BE PERFORMED. MUST CORRESPOND TO J FROM CARD F.7.A AND BE SUPPLIED IN ASCENDING ORDER.
MWSEG(2,J)	THE NUMBER OF THE CONTACT ELLIPSOID TO BE ASSOCIATED WITH SEGMENT NUMBER JJ.
MWSEG(3,J)	THE SEGMENT IDENTIFICATION NUMBER (NSEG+1 FOR THE VEHICLE, NSEG+2 FOR THE GROUND) ASSOCIATED WITH PLANE NUMBER MWSEG (4,J).
MWSEG(4,J)	THE PLANE IDENTIFICATION NUMBER FROM CARD D.2.A THROUGH WHICH IF SEGMENT J PASSES, WIND FORCE CALCULATIONS WILL BE PERFORMED.
MWSEG(5,J)	THE FUNCTION NUMBER FROM CARD E.6.A FOR THE WIND FORCE FUNCTION TO BE USED.

F.8 SUBROUTINE HINPUT - CARD INPUT FOR HARNESS-BELT SYSTEMS.

NOTE: NHRNSS WHICH WAS SUPPLIED ON CARD F.8.A FOR VERSION 12 IS NOW SUPPLIED ON CARD D.1. IF NHRNSS≠0, CARDS F.8 MUST BE SUPPLIED. PREVIOUSLY FOR VERSION 12, A BLANK CARD F.8.A WAS REQUIRED IF NO HARNESS BELT SYSTEMS WERE DESIRED.

```
CARD F.8.A                 FORMAT (5I4)

    NBLTPH(I),             NUMBER OF INDIVIDUAL BELTS FOR EACH HARNESS
     I=1,NHRNSS            NO. I. MAY BE ZERO OR BLANK. MAXIMUM VALUE
                           OF SUM OF ALL NBLTPH IS 20.

CARD F.8.A IS FOLLOWED BY NHRNSS SETS OF CARDS F.8.B - F.8.D.

CARD F.8.B                 FORMAT (18I4) USE TWO CARDS IF NBLTPH(I)>18.

    NPTSPB(J),             THE NUMBER OF REFERENCE POINTS INCLUDING
     J=1,NBLTPH(I)         ANCHOR POINTS FOR BELT NO. J OF HARNESS
                           NO. I. MAY BE ZERO OR BLANK. THE MAXIMUM
                           VALUE OF THE SUM OF ALL NPTSPB FOR ALL
                           HARNESS-BELT SYSTEMS IS 100. THE MAXIMUM
                           VALUE OF THE SUM OF ALL NPTSPB FOR ANY ONE
                           HARNESS BELT SYSTEM IS 50. THE MAXIMUM VALUE
                           OF ANY INDIVIDUAL NPTSPB IS 25.

EACH CARD F.8.B IS FOLLOWED BY NBLTPH(I) SETS OF CARDS F.8.C - F.8.D.

CARD F.8.C                 FORMAT (5I4, F12.0)

    NF(L),L=1,5            THE FUNCTION NUMBERS FROM CARDS E.1 TO DEFINE
                           THE STRESS-STRAIN OF BELT NO. J. THE DEFINITION
                           OF THESE FUNCTIONS ARE IDENTICAL TO THOSE OF
                           NF(1) TO NF(5) ON CARDS F.2.B, EXCEPT THAT THE
                           USE OF RATE DEPENDENT FUNCTIONS IS PERMITTED.

    XLONG(J)               THE INITIAL SLACK (IN) OF BELT NO. J. A NEG-
                           ATIVE VALUE CAN BE SPECIFIED TO INDICATE A
                           PRE-TIGHTENED BELT. THE PROGRAM WILL ADD THIS
                           TO THE INITIAL GEOMETRIC LENGTH TO OBTAIN THE
                           INITIAL BELT LENGTH AND DISTRIBUTE THE SLACK
                           PROPORTIONATELY BETWEEN THE POINTS.
```

```
EACH CARD F.8.C IS FOLLOWED BY NPTSPB(J) PAIRS OF F.8.D1 AND D2 CARDS     S
TO SPECIFY THE REFERENCE POINTS (K) FOR BELT (J) OF HARNESS (I).          S

CARD F.8.D1              FORMAT (9I4, 3F12.0)                               S

     KS                  INTEGER OF THE FORM 100*KTP+KSEG, WHERE KSEG       S
                         IS THE IDENTIFICATION NUMBER OF THE SEGMENT        *
                         ASSOCIATED WITH REFERENCE POINT (K), AND           *
                         KTP IS A TIE-POINT IDENTIFICATION NUMBER           S
                         WHICH MAY BE BLANK OR ZERO. ALL POINTS (K)         S
                         OF HARNESS (I) THAT HAVE THE SAME NON-ZERO         S
                         VALUE FOR KTP (THERE SHOULD BE ONLY ONE FOR        S
                         EACH BELT (J)) WILL BE CONNECTED  AND SHOULD       S
                         HAVE IDENTICAL VALUES FOR ALL OTHER INPUT.         S

     KE                  THE IDENTIFICATION NUMBER OF THE CONTACT           *
                         ELLIPSOID ASSOCIATED WITH REFERENCE POINT          *
                         NO. K. IF NO ELLIPSOID IS SPECIFIED (KE=0),        *
                         THE PROGRAM WILL ASSUME A UNIT SPHERE.             *

     NPD                 INDICATOR FOR THE PREFERRED DIRECTION OPTION.      S
                         IF A NON-ZERO INTEGER IS GIVEN, A NON-ZERO         S
                         VECTOR MUST BE SPECIFIED FOR BAR(L,K),L=10,12      S
                         ON CARD F.8.D2. THE REFERENCE POINT WILL BE        S
                         ALLOWED TO MOVE ALONG THE SURFACE IN A DIRECTION   S
                         WHICH IS PERPENDICULAR BOTH TO THIS VECTOR AND     S
                         TO THE NORMAL OF THE SURFACE SUBJECT TO THE        S
                         CONSTRAINT IMPOSED BY D2 OF FUNCTION NF(5)         S
                         BELOW. IF NPD=0, THE NOMINAL BELT LINE IS USED     S
                         IN PLACE OF THIS VECTOR. NPD MUST BE NONZERO       S
                         IF POINT NO. K IS A TIE PONT.                      S

     NDR                 INDICATOR FOR THE DELTA R OPTION. IF NDR = 0,      S
                         BELT (J) WILL BE ALLOWED TO SLIP AT REFERENCE      S
                         POINT (K). IF NDR ≠ 0, BELT (J) WILL NOT SLIP      S
                         BUT REFERENCE POINT (K) WILL BE MOVED ALONG THE    S
                         NOMINAL BELT LINE. IN BOTH CASES THE SLIPPAGE      S
                         OR MOTION IS SUBJECT TO THE CONSTRAINT IMPOSED     S
                         BY THE COEFFICIENT OF FRICTION GIVEN BY D4 OF      S
                         FUNCTION NF(5) BELOW. NDR MUST BE NON-ZERO         S
                         FOR END REFERENCE POINTS OF THE BELT.              S

     NF(L),L=1,4         THE FUNCTION NUMBERS FROM CARDS E.1 TO DEFINE      S
                         THE FORCE DEFLECTION FUNCTION BETWEEN BELT (J)     S
                         AND REFERENCE POINT (K). IF NF(1) = 0, THE         S
                         SURFACE IS TREATED AS RIGID AND NO PERTURBATION    S
                         OF THE REFERENCE POINT NORMAL TO THE SURFACE       S
                         IS ALLOWED. THE USE OF RATE DEPENDENT FUNCT-       S
                         IONS AS DEFINED UNDER CARDS F.1.B IS PERMITTED.    S
```

NF(5)	THE FUNCTION NUMBER FROM CARD E.1 TO DEFINE THE FRICTION COEFFICIENTS FOR BELT (J) AT REFERENCE POINT (K). TWO CONSTANT VALUES ARE TO BE DEFINED ON CARD E.2 OF THIS FUNCTION BY SETTING DD = D1 = D3 = 0. D2 IS THE COEFFICIENT OF FRICTION PERPENDICULAR TO THE NOMINAL BELT LINE ALONG THE SURFACE AND D4 IS THE COEFFICIENT OF FRICTION ALONG THE NOMINAL BELT LINE. IF NF(5) = 0, INFINITE FRICTION IS ASSUMED.	S
BAR(L,K),L=1,3	THE X,Y AND Z COORDINATES (IN) OF REFERENCE POINT (K) OF BELT (J) IN THE LOCAL COORDINATE SYSTEM OF SEGMENT NO. KS. IF AN ELLIPSOID IS SPECIFIED (KE≠0), THE POINT IS REFERRED TO THE CENTER OF THE ELLIPSOID AND THE SUPPLIED VALUES WILL BE ADJUSTED BY THE PROGRAM TO LIE ON THE ELLIPSOID SURFACE. IF KE = 0, A NON-ZERO VECTOR MUST BE SPECIFIED. THIS VECTOR WILL BE USED TO TO COMPUTE THE NORMAL IN THE DEFINITION OF ITS LOCAL COORDINATE SYSTEM AND TO RESOLVE THE BELT FORCES. THE PROGRAM WILL ASSUME THAT BELT (J) WILL RUN THROUGH THE POINTS IN THE SPECIFIED ORDER. HOWEVER, IF THE FORCES ARE SUCH AS TO PULL THE BELT AWAY FROM THE SURFACE, THIS POINT WILL BE IGNORED IF IT IS NOT AN END OR ATTACHMENT POINT.	*

CARD F.8.02	FORMAT (6F12.0)	S
BAR(L,K),L=7,9	THE X,Y AND Z COORDINATES (IN) OF THE OFFSET IN THE LOCAL COORDINATE SYSTEM OF SEGMENT KS. THIS VECTOR IS ADDED TO THE REFERENCE VECTOR DEFINED ABOVE (L=1,3) TO DETERMINE THE LOCATION OF THE REFERENCE POINT (K) RELATIVE TO THE C.G. OF SEGMENT KS.	S
BAR(L,K),L=10,12	THE X,Y AND Z COORDINATES OF A VECTOR IN THE LOCAL COORDINATE SYSTEM OF SEGMENT KS. THIS VECTOR IS USED FOR THE PREFERRED DIRECTION (SEE NPD ABOVE). THIS VECTOR MUST NOT BE PARALLEL TO THE NORMAL COMPUTED FROM BAR(L,K), FOR L=1,3 ABOVE.	S

G. SUBROUTINE INITAL

CARD G.1.A — FORMAT (3F10.0, 5I4)

ZPLT(I),I=1,3	THE X, Y, AND Z PLOT COORDINATES (FOR SUBROUTINE PRIPLT) OF THE ORIGIN OF THE VEHICLE REFERENCE SYSTEM. $0 < X < 61$, $0 < Y < 61$, $0 < Z < 121$
I1	A VALUE OF 15 IS REQUIRED TO CALL SUBROUTINE EQUILB AND PROCESS CARDS G.4, G.5 AND G.6.
J1	IF NON-ZERO, CARD G.1.B IS REQUIRED TO DEFINE SCALING INFORMATION FOR THE PRINTER PLOTS
I2,J2	CURRENTLY NOT USED BY THE PROGRAM.
I3	IF ZERO, SEGMENT AND ANGULAR VELOCITIES ARE NOT SUPPLIED ON THE FOLLOWING CARDS BUT ARE SET EQUAL TO THE INITIAL VEHICLE VELOCITY. IF I3 # 0, SEGLV AND WMGDEG MUST BE SUPPLIED.

IF J1 IS ZERO OR BLANK ON CARD G.1.A, THE FOLLOWING CARD G.1.B SHOULD NOT BE SUPPLIED AND DEFAULT VALUES OF 10.0, 6.0 AND 1.0 WILL BE USED FOR THE SPLT ARRAY.

CARD G.1.B — FORMAT (3F10.0)

SPLT(1)	THE NUMBER OF HORIZONTAL PRINT POSITIONS PER UNIT LENGTH FOR THE OUTPUT UNIT THAT WILL PRINT THE PRINTER PLOTS PRODUCED BY SUBROUTINE PRIPLT (NORMAL VALUE IS 10.0 FOR 10 SPACES OR COLUMNS PER INCH).
SPLT(2)	THE NUMBER OF VERTICAL PRINT LINES PER UNIT LENGTH (NORMAL VALUES ARE 6.0 OR 8.0 FOR 6 OR 8 LINES PER INCH). THE PROGRAM USES ONLY THE RATIO OF SPLT(1) TO SPLT(2).
SPLT(3)	SCALE FACTOR THAT REPRESENTS THE DISTANCE (INCHES OR LENGTH UNIT ON CARD A.3) BETWEEN VERTICAL PRINT LINES FOR THE PRINTER PLOTS. NOTE: THE PRINTER PLOT WAS ORIGINALLY DESIGNED FOR 120X60 UNITS (INCHES) ALONG THE Z AND X OR Y DIRECTIONS WHICH MAY NOT BE SATISFACTORY FOR CERTAIN SITUATIONS (E.G., METRIC UNITS).

ONE G.2 CARD MUST BE SUPPLIED FOR EACH REFERENCE SEGMENT (I.E., SEGMENT NO. 1 AND FOR EACH SEGMENT J+1 WHERE JNT(J) = 0 ON CARDS B.3) IN ASCENDING SEGMENT NUMBER SEQUENCE.

```
CARDS G.2.A - G.2.M        FORMAT (6F10.0)

    SEGLP(I,J),I=1,3       THE INITIAL X, Y, AND Z COORDINATES OF THE
                           JTH BODY SEGMENT IN INERTIAL REFERENCE (IN).

    SEGLV(I,J),I=1,3       THE INITIAL X, Y, AND Z COMPONENTS OF VELOCITY
                           OF THE JTH BODY SEGMENT IN INERTIAL REFER-
                           ENCE (IN/SEC). THESE FIELDS MAY BE LEFT BLANK
                           IF I3 = 0 ON CARD G.1 IN WHICH CASE THE
                           INITIAL VELOCITY OF THE VEHICLE WILL BE USED.


CARDS G.3.A1-G.3.N1        FORMAT (6F10.0, 4I3)                              *
 (NSEG CARDS OR SETS OF G.3.J1,G.3.J2 CARDS)                                 *

    YPR(I,J),I=1,3         THE INITIAL ROTATION ANGLES (DEGREES) OF THE      *
                           JTH SEGMENT ABOUT THE LOCAL Z, Y AND X AXES       *
                           OF THE SEGMENT GIVEN BY ID(I,J) IN THE ORDER      *
                           SPECIFIED BY ID(I,J),I=1,3 BELOW.                 *

    WMGDEG(I,J),I=1,3      THE INITIAL COMPONENTS OF ANGULAR VELOCITY
                           ABOUT THE LOCAL X,Y AND Z AXES OF THE JTH
                           BODY SEGMENT (DEG/SEC). IF I3 = 0 ON CARD
                           G.1, THE INITIAL ANGULAR VELOCITY OF THE
                           VEHICLE WILL BE CONVERTED TO THE SEGMENT
                           REFERENCE AND WILL BE USED.

    ID(I,J),I=1,3          INDICATORS USED TO SPECIFY THE ORDER OF THE       *
                           AXES OF THE ROTATIONS GIVEN IN YPR ABOVE.         *
                           (SEE COMPLETE DEFINITION UNDER CARDS B.3.A2.)     *
                           ZEROS OR BLANKS WILL DEFAULT TO 1,2 AND 3 TO      |
                           INDICATE THAT THE STANDARD SEQUENCE OF YAW,       |
                           PITCH AND ROLL IS REVERSED (AS REQUIRED BY        |
                           VERSIONS PREVIOUS TO 18A OF THE PROGRAM).         |

                           VALUES OF 3,2,1 INDICATES THAT THE STANDARD       *
                           YAW, PITCH AND ROLL SEQUENCE BE USED.             *

                           VALUES OF 3,1,-3 INDICATES THAT PRECESSION,       *
                           NOTATION AND SPIN FOR EULER JOINTS BE USED.       *

                           A NEGATIVE VALUE FOR ID(1,J) INDICATES THAT       *
                           PROJECTIONS OR PROJECTION ANGLES OF THE           *
                           PRINCIPAL AXES OF SEGMENT J WILL BE USED AND      *
                           THAT A CARD G.3.J2 WILL FOLLOW THIS CARD.         *
```

```
ID(4,J)              THE SEGMENT NUMBER TO WHICH THE ROTATIONS         *
                     GIVEN BY YPR OR BY ANGLES ON CARD G.3.J2          *
                     ARE RESPECT TO. A VALUE OF ZERO OR BLANK WILL     *
                     DEFAULT TO THE GROUND (NSEG+NBAG+2) OR INERTIAL   *
                     REFERENCE. THE VEHICLE MAY BE SPECIFIED BY        *
                     SUPPLYING NSEG+1. OTHERWISE THE NO. OF THE        *
                     SEGMENT MUST BE LESS THAN J. A NEGATIVE NUMBER    *
                     (-|JNT(J-1)|, AS SPECIFIED ON CARD B.3.A1)        *
                     MAY BE USED TO DEFINE THE ROTATION ANGLES         *
                     WITH RESPECT TO THE JOINT PRINCIPAL AXES AS       *
                     SPECIFIED ON CARD B.3.A2.                         *

NOTE: THE VALUES OF YPR AND ID ARE USED TO COMPUTE A DIRECTION COSINE  |
MATRIX R. THE DIRECTION COSINE MATRIX D(J) OF SEGMENT J IS DETERMINED  |
BY THE VALUE OF K = ID(4,J) AS FOLLOWS:                                |
                                                                       |
  K = 0: D(J) = R(J)                   ( K=0 OR EQUAL TO NGRND )       |
  K > 0: D(J) = R(J)D(K)               ( K<J OR EQUAL TO NVEH )        |
  K < 0: D(J) = H'(J)R(J)H(K)D(K)      ( K = -|JNT(J-1)| )             |
                                                                       |
THERE ARE NO RESTRICTIONS ON A BALL OR EULER JOINT. AN EULER JOINT     |
CAN BE SET TO AN INITIAL PRECESSION(P), NUTATION(N) AND SPIN(S) BY     |
SPECIFYING YPR = P,N,S AND ID = 3,1,-3,-|JNT(J-1)|. TO PRESERVE THE    |
AXES OF A PIN JOINT, CARE MUST BE TAKEN THAT THE RELATIVE ORIENTATION  |
OF SEGMENTS J AND JNT(J-1) REPRESENTS A ROTATION ABOUT THE PIN AXIS    |
ONLY. (THE PIN AXIS IS ALWAYS THE Y AXIS OF THE JOINT PRINCIPAL AXES   |
AS SPECIFIED ON CARD B.3.A2.) THIS CAN BE ASSURED BY SUPPLYING YPR =   |
0,P,0 AND ID = 0,0,0,-|JNT(J-1)| WHERE P IS THE PITCH OF SEGMENT J     |
WITH RESPECT TO THE CENTER OF SYMMETRY (CARD B.3.A2) OF JOINT J-1.     |
FOR THE CASE WHERE THE Y AXES OF SEGMENTS J AND JNT(J-1) ARE PARALLEL  |
TO THE PIN AXIS, THE PIN AXIS CAN BE PRESERVED BY SUPPLYING VALUES     |
OF YPR = 0,P,0 AND ID = 0,0,0,+|JNT(J-1)| WHERE P IS THE PITCH OF      |
SEGMENT J WITH RESPECT TO SEGMENT JNT(J-1).                            |
```

```
A CARD G.3.J2 MUST FOLLOW ANY CARD G.3.J1 ON WHICH ID(1,J) IS NEGATIVE.  *

CARDS G.3.A2-G.3.N2      FORMAT (6F10.0, 4I3)                             *

    A1,A2,A3             SPECIFIES THE PROJECTION OF THE PRIMARY AXIS     *
                         GIVEN BY IK BELOW. IF II IS NEGATIVE, VALUES     *
                         WILL BE THE X,Y AND Z COMPONENTS (IN) IN THE     *
                         PROJECTION REFERENCE SYSTEM OF A VECTOR          *
                         ALONG THE POSITIVE IK AXIS OF SEGMENT NO. J.     *
                         IF II IS POSITIVE, A1,A2 (A3 NOT USED) ARE THE   *
                         PROJECTION ANGLES (DEG) OF THE POSITIVE IK AXIS  *
                         OF SEGMENT NUMBER J IN TWO OF THE PROJECTION     *
                         REFERENCE PLANES SPECIFIED BY THE VALUE OF II.   *

    B1,B2,B3             SPECIFIES THE PROJECTION OF A SECONDARY AXIS     *
                         GIVEN BY JK BELOW. DEFINITION IS IDENTICAL TO    *
                         A1,A2,A3 ABOVE BUT USES JJ AND JK INSTEAD        *
                         OF II AND IK.                                    *

    II                   IF II IS NEGATIVE, THE COMPONENTS OF A VECTOR    *
                         ALONG THE POSITIVE IK AXIS WILL BE GIVEN BY A1,  *
                         A2,A3. IF II IS POSITIVE, A VALUE OF 1,2 OR 3    *
                         IS USED TO INDICATE THAT THE X,Y OR Z AXIS IS    *
                         THE COMMON AXIS OF THE TWO PROJECTION REFERENCE  *
                         PLANES USED TO SPECIFY THE TWO PROJECTION        *
                         ANGLES AS FOLLOWS:                               *

                         IF II=1, A1 IN Z-X PLANE, A2 IN X-Y PLANE.       *
                         IF II=2, A1 IN X-Y PLANE, A2 IN Y-Z PLANE.       *
                         IF II=3, A1 IN Y-Z PLANE, A2 IN Z-X PLANE.       *

                         IN THE X-Y PLANE, THE ANGLE IS MEASURED FROM     *
                         THE X-AXIS, POSITIVE TOWARD THE Y AXIS.          *

                         IN THE Y-Z PLANE, THE ANGLE IS MEASURED FROM     *
                         THE Y-AXIS, POSITIVE TOWARD THE Z AXIS.          *

                         IN THE Z-X PLANE, THE ANGLE IS MEASURED FROM     *
                         THE Z AXIS, POSITIVE TOWARD THE X AXIS.          *

                         RESTRICTION: SIN(A1) * COS(A2) CANNOT BE ZERO.   *

    IK                   A VALUE OF 1,2 OR 3 TO SPECIFY THAT THE X,Y      *
                         OR Z AXIS OF SEGMENT NUMBER J IS THE PRIMARY     *
                         AXIS TO BE PROJECTED.                            *

    JJ,JK                SAME DEFINITION AS FOR II,IK ABOVE BUT FOR A     *
                         SECONDARY AXIS OF SEGMENT NUMBER J. THE VALUE    *
                         OF JK MUST BE DIFFERENT THAN THAT OF IK.          *
```

```
SUBROUTINE EQUILB                                                     *
                                                                      *
CARDS G.4, G.5 AND G.6 ARE REQUIRED IF I1 = 15 ON CARD G.1.           *
                                                                      *
CARD G.4                FORMAT (2I4)                                  *
                                                                      *
    NVAR                NO. OF INDEPENDENT VARIABLES SUPPLIED ON CARDS *
                        G.2 AND G.3 THAT ARE TO BE ADJUSTED SUCH THAT  *
                        CONTACT NORMAL FORCES ARE EQUAL TO EITHER GX   *
                        SUPPLIED ON CARDS G.5 OR CONSTRAINT NORMAL     *
                        FORCES CONTROLLED BY CARDS G.6 (MAX = 10).     *
                                                                      *
    NCON                NO. OF CONSTRAINTS TO BE IMPOSED TO COMPUTE    *
                        THOSE CONSTRAINT FORCES WHICH WILL BE SATISFIED *
                        BY INITIAL CONTACT FORCES. IF ZERO, THE SUPPLIED *
                        VALUES OF GX WILL BE USED. (MAX = 5)           *
                                                                      *
                                                                      *
CARDS G.5.A - G.5.N     FORMAT (3I4, 2F8.0, 8I4)                      *
     (NVAR CARDS)                                                     *
                                                                      *
    NTV(J)              INDICATES TYPE OF JTH INDEPENDENT VARIABLE     *
                          1 - SEGLP FROM CARDS G.2                     *
                          2 -   YPR FROM CARDS G.3                     *
                                                                      *
    NII(J)              A VALUE OF 1,2 OR 3 TO INDICATE THE X,Y OR Z   *
                        COORDINATE OF SEGLP IF NTV(J)=1, OR YAW, PITCH *
                        OR ROLL OF YPR IF NTV(J)=2.                    *
                                                                      *
    NSG(J)              THE SEGMENT NUMBER (AS SPECIFIED BY INDEX I    *
                        OF CARDS B.2) FOR THE JTH INDEPENDENT VARIABLE. *
                                                                      *
    GX(J)               THE MAGNITUDE OF THE CONTACT NORMAL FORCE FOR  *
                        THE JTH INDEPENDENT VARIABLE (LBS.). IF THIS   *
                        CONTACT IS TO BE CONTROLLED BY A CONSTRAINT ON *
                        CARDS G.6 (I.E., J=INDGX(I)), THE SUPPLIED     *
                        VALUE OF GX WILL BE THE INITIAL VALUE FOR THE  *
                        ITERATION OF THE CONTACT NORMAL FORCE TO EQUAL *
                        THE CONSTRAINT NORMAL FORCE; OTHERWISE, THE JTH *
                        INDEPENDENT VARIABLE WILL BE ADJUSTED SUCH THAT *
                        THE CONTACT NORMAL FORCE WILL BE EQUAL TO GX.  *
                                                                      *
    XDEV(J)             THE MAXIMUM ALLOWABLE DEVIATION FROM THE INITIAL *
                        POSITIONS SPECIFIED ON CARDS G.2 AND G.3 DURING *
                        THE ITERATION OF THE JTH INDEPENDENT VARIABLE  *
                        FOR THE CONTACT NORMAL FORCE TO EQUAL GX. IF   *
                        EXCEEDED, THE PROGRAM WILL TERMINATE WITH AN   *
                        ERROR MESSAGE.  IF XDEV = 0, THE TESTS WILL    *
                        NOT BE PERFORMED.                              *
                                                                      *
```

JPL(J)	THE PLANE NUMBER CORRESPONDING TO NJ ON CARDS F.1.B - F.1.N FOR THE CONTACT WHOSE NORMAL FORCE IS TO BE CONTROLLED BY THE JTH VARIABLE.
JSG(J)	THE SEGMENT IDENTIFICATION NUMBER (AS SPECIFIED BY INDEX I OF CARDS B.2) INVOLVED IN THE CONTACT WITH PLANE NO. JPL(J). NOTE: A CONTACT FOR THIS PLANE AND SEGMENT MUST HAVE BEEN SET UP ON CARDS F.1.B - F.1.N.
NAV(J)	NO. OF VARIABLES ASSOCIATED WITH THE JTH INDEPENDENT VARIABLE. (MAX= 5, MAY BE ZERO)
KSG(I,J),I=1,NAV	THE SEGMENT NUMBERS (DEFINITION SAME AS FOR NSG(J)) FOR THE NAV(J) VARIABLES ASSOCIATED WITH THE JTH INDEPENT VARIABLE. ANY CHANGE MADE TO THE JTH INDEPENDENT VARIABLE TO ACHIEVE INITIAL EQUILIBRIUM WILL ALSO BE MADE TO THE CORRESPONDING VARIABLES FOR THESE SEGMENTS SUCH THAT THE INITIAL RELATIVE ORIENTATION WILL BE MAINTAINED AS SPECIFIED ON CARDS G.2 AND G.3.
CARDS G.6.A - G.6.M (NCON CARDS)	FORMAT (4I4)
IPL(I),ISG(I)	THE PLANE AND SEGMENT NUMBERS (DEFINITION SAME AS FOR JPL(J) AND JSG(J) ABOVE) FOR THE ITH CONSTRAINT TO BE IMPOSED FOR INITIAL EQUILIBRIUM DURING THE CONTACT NORMAL FORCE TO CONSTRAINT NORMAL FORCE ITERATION.
LTYPE(I)	INDICATES THE TYPE OF THE ITH CONSTRAINT 3 - ROLL CONSTRAINT 4 - SLIDE CONSTRAINT
INDGX(I)	THE INDEX J (FROM 1 TO NVAR) FROM CARD G.5 FOR WHOSE CONTACT NORMAL FORCE WILL BE ITERATED TO BE EQUAL TO THE ITH CONSTRAINT NORMAL FORCE. MAY BE ZERO, BUT IF INDGX(I) = J, THEN IPL(I) AND ISG(I) MUST BE EQUAL TO JPL(J) AND JSG(J).

NOTE: SUBROUTINE EQUILB WILL ADJUST THE INITIAL POSITION PARAMETERS SUPPLIED ON CARDS G.2 AND G.3. IF THE CONSTRAINTS TEMPORARILY IMPOSED BY CARDS G.6 PROPERLY CONSTRAIN ALL OF THE SEGMENTS, ZERO ACCELERATIONS WILL BE OBTAINED WHILE THE CONSTRAINTS ARE ON. THE ITERATION WILL PRODUCE NORMAL AND TANGENTIAL CONTACT FORCES THAT WILL RESULT IN SMALL (< 0.02 G) INITIAL LINEAR ACCELERATIONS FOR ALL OF THE BODY SEGMENTS. FOR THE SEATED "STANDARD" FIFTEEN SEGMENT OCCUPANT, THIS CAN BE ACHIEVED AS FOLLOWS:

A. LOCK JOINT P, W, NP, HP, RA AND LA BY SETTING IPIN = -2 ON CARDS B.3. IF THE MAXIMUM TORQUE FOR A LOCKED JOINT (T1 FOR VISC(4,3*J-2) ON CARDS B.5) IS ZERO, THEN SUBROUTINE EQUILB WILL SET T1 FOR THESE LOCKED JOINTS TO 1.5 TIMES THE MAGNITUDE OF THE JOINT TORQUE FINALLY PRODUCED AT TIME ZERO.

B. CONSTRAIN THE ARMS BY EITHER SETTING UP FIXED POINT CONSTRAINTS (TYPE=1) FOR THE RLA AND LLA WITH THE VEHICLE ON CARDS D.6, OR LOCK THE JOINTS RS, RE, LS AND LE AS IN STEP A ABOVE. IF THE CONSTRAINTS ARE IMPOSED ON CARDS D.6, SUBROUTINE EQUILB WILL ADJUST THE POINT ON THE VEHICLE (RK2 ON CARDS D.6) FOR ANY TYPE 1 CONSTRAINT INVOLVING THE VEHICLE SO THAT IT WILL COINCIDE WITH THE SPECIFIED POINT ON THE BODY SEGMENT (RK1 ON CARDS D.6) AS ADJUSTMENTS ARE MADE TO THE INITIAL POSITION PARAMETERS.

C. SET UP ALLOWED CONTACTS AND ASSOCIATED FORCE DEFLECTION FUNCTIONS ON CARDS F.1 FOR THE SEAT CUSHION PLANE WITH THE LT, RUL AND LUL SEGMENTS, THE SEAT BACK PLANE WITH THE LT, CT AND UT SEGMENTS, AND THE FLOORBOARD PLANE WITH THE RF AND LF SEGMENTS.

D. SET UP INITIAL POSITION PARAMETERS ON CARDS G.2 AND G.3 THAT ARE JUST "SHORT OF" OR CLOSE TO THE FINAL PENETRATION DISTANCES FOR THE SEGMENTS WITH THE CONTACT PLANES.

E. SET NVAR = 5 AND NCON = 4 ON CARD G.4.

F. SUPPLY THE FOLLOWING INPUT PARAMETERS ON CARDS G.5:

J	NTV	NI1	NSG	GX	XDEV	JPL	JSG	NAV	KSG
1	1	3	(LT)	90.0	1.0	(SEAT CUSHION)	(LT)	0	
2	1	1	(LT)	5.0	1.0	(SEAT BACK)	(LT)	0	
3	2	2	(UT)	10.0	5.0	(SEAT BACK)	(UT)	4	(LT),(CT),(N),(H)
4	2	2	(RUL)	25.0	10.0	(SEAT CUSHION)	(RUL)	1	(LUL)
5	2	2	(RLL)	10.0	10.0	(FLOORBOARD)	(RF)	1	(LLL)

( ) INDICATES THAT IDENTIFICATION NUMBER SHOULD BE USED

G. SUPPLY THE FOLLOWING INPUT PARAMETERS ON CARDS G.6:

I	IPL	ISG	LTYPE	INDGX
1	(SEAT CUSHION)	(LT)	3	1
2	(SEAT BACK)	(UT)	4	3
3	(FLOORBOARD)	(RF)	3	5
4	(FLOORBOARD)	(LF)	3	0

USING THE ABOVE INPUT PARAMETERS, SUBROUTINE EQUILB WILL ADJUST THE X AND Z COORDINATES OF THE LT, THE PITCH ANGLES (MAINTAINING THE INITIAL RELATIVE ORIENTATION) OF THE UT, LT, CT, N AND H SEGMENTS, THE RUL AND LUL SEGMENTS, AND THE RLL AND LLL SEGMENTS, AND THE INITIAL NORMAL CONTACT FORCES (GX) OF THE SEAT CUSHION WITH THE LT, THE SEAT BACK WITH THE UT AND THE FLOORBOARD WITH THE RF. IT IS BELIEVED THAT THE RESULTING INITIAL POSITIONS ARE UNIQUE AND ARE FUNCTIONS OF THE VALUES OF THE CONTACT NORMAL FORCES (GX) SUPPLIED FOR THE SEAT BACK WITH THE LT AND THE SEAT CUSHION WITH THE RUL CONTACTS.

H. SUBROUTINE OUTPUT

THIS SUBROUTINE PROVIDES INPUT TO CONTROL THE DESIRED TIME HISTORY OUTPUT OF SELECTED SEGMENT LINEAR AND ANGULAR ACCELERATIONS, VELOCITIES, AND DISPLACEMENTS, AND JOINT PARAMETERS.

H.1 (K=1) SEGMENT LINEAR ACCELERATIONS IN LOCAL REFERENCE

CARD H.1.A — FORMAT (2I6, 3F12.6)

NSG(K)	THE NUMBER OF SELECTED POINTS ON THE VARIOUS BODY SEGMENTS FOR WHICH TIME HISTORIES ARE DESIRED. THE MAXIMUM VALUE FOR NSG(K) IS 20. IF NSG(K) IS 0, INSERT 2 BLANK CARDS. IF NSG(K) IS 1, A SINGLE BLANK CARD SHOULD FOLLOW CARD H.1.K.
MSG(1,K)	THE SEGMENT NUMBER OF THE FIRST POINT AS DETERMINED BY THE INDEX I ON CARDS B.2.A - B.2.N. THE VEHICLE MAY BE SPECIFIED BY NSEG+1, OR THE JTH AIRBAG BY NSEG+1+J. *
XSG(I,1,K),I=1,3	THE X, Y, AND Z COORDINATES IN SEGMENT REFERENCE OF THE FIRST POINT (INCHES).

FOLLOWED BY NSG(K)-1 CARDS OF THE FOLLOWING (J = 2, NSG(K))

CARDS H.1.B - H.1.N — FORMAT (I12, 3F12.6)

MSG(J,K)	SAME AS ABOVE BUT FOR THE JTH POINT.
XSG(I,J,K),I=1,3	SAME AS ABOVE BUT FOR THE JTH POINT.

H.2 (K=2) SEGMENT LINEAR VELOCITIES IN VEHICLE REFERENCE

CARDS H.2.A - H.2.N — FORMAT (2I6, 3F12.6/ (I12, 3F12.6))

DESCRIPTION SAME AS FOR H.1.

H.3 (K=3) SEGMENT LINEAR DISPLACEMENTS IN VEHICLE REFERENCE

CARDS H.3.A - H.3.N — FORMAT (2I6, 3F12.6/ (I12, 3F12.6))

DESCRIPTION SAME AS FOR H.1.

H.4 (K=4) SEGMENT ANGULAR ACCELERATIONS IN LOCAL REFERENCE

CARD H.4 — FORMAT (12I6/ (I12, 10I6))

NSG(K)	THE NUMBER OF SELECTED SEGMENTS FOR WHICH TIME HISTORIES ARE DESIRED (MAXIMUM = 20). SUPPLY BLANK CARD IF NONE ARE DESIRED.
MSG(J,K),J=1,KSG WHERE KSG=NSG(K)	THE SEGMENT NUMBERS AS DETERMINED BY INDEX I ON CARDS B.2.A - B.2.N. THE VEHICLE MAY BE SPECIFIED BY NSEG+1, * OR THE JTH AIRBAG BY NSEG+1+J. * IF NSG(K) > 11, USE THE SECOND CARD, LEAVING THE FIRST FIELD OF 6 COLUMNS BLANK. IF NSG(K) = 11, A SECOND CARD, COMPLETELY BLANK, SHOULD FOLLOW THIS CARD.

H.5 (K=5) SEGMENT ANGULAR VELOCITIES IN VEHICLE REFERENCE

CARD H.5 — FORMAT (12I6/ (I12, 10I6))

DESCRIPTION SAME AS FOR H.4.

H.6 (K=6) SEGMENT ANGULAR DISPLACEMENTS IN VEHICLE REFERENCE

CARD H.6 — FORMAT (12I6/ (I12, 10I6))

DESCRIPTION SAME AS FOR H.4.

H.7 (K=7) JOINT PARAMETERS

CARD H.7 — FORMAT (12I6/ (I12, 10I6))

NSG(K)	THE NUMBER OF SELECTED JOINTS FOR WHICH TIME HISTORIES ARE DESIRED. INSERT BLANK CARD IF NONE ARE DESIRED (NJNT MAXIMUM).
MSG(J,K),J=1,KSG WHERE KSG=NSG(K)	THE JOINT NUMBERS AS DETERMINED BY INDEX J ON CARDS B.3.A - B.3.J. IF NSG(K) > 11, USE A SECOND CARD LEAVING THE FIRST FIELD OF 6 COLUMNS BLANK. IF NSG(K) = 11, A SECOND CARD, COMPLETELY BLANK, SHOULD FOLLOW THIS CARD.

H.8 (SUBROUTINE POSTPR) - HIC, HSI AND CSI CALCULATIONS. *
THIS CARD IS REQUIRED WHENEVER SUBROUTINE POSTPR IS CALLED AS DETER- *
MINED BY THE VALUE OF NPRT(4) ON CARD A.5 (ALL VALUES BUT 0 OR 4). *

CARD H.8 FORMAT (18I4) *

JDTPTS(1)	THE INDEX J ON CARDS H.1 CORRESPONDING TO THE HEAD C.G. WHOSE RESULTANT ACCELERATION TIME HISTORY WILL BE USED TO COMPUTE THE HEAD INJURY CRITERIA (HIC) AND HEAD SEVERITY INDEX (HSI). THE COMPUTATIONS WILL NOT BE DONE IF JDTPTS(1) = 0 OR BLANK.	*
JDTPTS(2)	THE INDEX J ON CARDS H.1 CORRESPONDING TO THE POINT WHOSE RESULTANT ACCELERATION TIME HISTORY WILL BE USED TO COMPUTE THE CHEST SEVERITY INDEX (CSI). THE COMPUTATIONS WILL NOT BE DONE IF JDTPTS(2) = 0 OR BLANK.	*

I. SUBROUTINE POSTPR

CARDS I ARE REQUIRED ONLY IF NPRT(4) IS AN ODD INTEGER ON CARD A.5. (SEE NOTE IN SUBROUTINE SLPLOT REGARDING PROGRAM CHANGES THAT MAY BE NECESSARY ON PLOTTING FACILITIES OTHER THAN THOSE AT CALSPAN.)

THESE CARDS ESSENTIALLY SPECIFY ALL OF THE ARGUMENTS TO SUBROUTINE SLPLOT AND THE INDICEES OF THE DATA IN THE TABULAR TIME HISTORIES TO BE PLOTTED. THE ABILITY EXISTS TO PLOT ANY SET OF VARIABLES IN THE TIME HISTORIES AS A FUNCTION OF ANY OTHER VARIABLE ON A FIXED (SPECIFIED BY THE USER INPUT) X-Y AXIS. BOTH AXES MAY BE EITHER LINEAR OR LOGARITHMIC. ANY DATA FALLING OUTSIDE OF THE SPECIFIED RANGE OF EACH AXIS WILL BE IGNORED. THE INPUT ALSO SPECIFIES THE X AND Y AXIS LABELS AND TWO LINES OF PLOT IDENTIFICATION THAT LIES BELOW THE X AXIS LABEL.

CARD I.1	FORMAT (18I4)
NPLT	THE NUMBER OF PLOTS TO BE GENERATED (MAX=20). (IF NPLT > 17, USE TWO CARDS.)
NYP(K),K=1,NPLT	THE NUMBER OF Y VARIABLES TO BE PLOTTED VS. THE SAME X VARIABLE FOR EACH OF THE NPLT PLOTS. NPLT + SUM OF NYP IS LIMITED TO 25.

A SET OF CARDS I.2-I.8 IS REQUIRED FOR EACH OF THE NPLT PLOTS.

CARD I.2.K	FORMAT (18I4)
MX1(K),MX2(K)	THE PAGE NO. (MX1) AND COLUMN NO. (MX2) FROM THE TABULATED TIME HISTORIES OF THE X (HORIZONTAL) VARIABLE FOR THE KTH PLOT. THESE PAGE NOS. START WITH 21 SO MX1 > 20. MX2 = 0 REFERS TO TIME (MSEC), THE LEFTMOST COLUMN. MX2 CAN BE SUPPLIED AS A NEGATIVE INTEGER TO INDICATE THAT THE VALUE FOR TIME ZERO WILL BE SUBTRACTED FROM ALL VALUES FOR PLOTTING PURPOSES.
MY1(J,K),MY2(J,K) FOR J=1,NYP(K)	THE PAGE NO. (MY1) AND COLUMN NO. (MY2) FOR THE NYP(K) Y (VERTICAL) VARIABLES TO BE PLOTTED VS. THE X VARIABLE SPECIFED BY MX1 AND MX2 FOR THE KTH PLOT. DEFINITION OF EACH MY1,MY2 SAME AS FOR MX1,MX2 ABOVE.

```
CARD I.3.K                  FORMAT (I4, 4X, 4F8.0)                       *
                                                                         *
    NX(K)                   THE NUMBER OF INTERVALS OR PLOTTING DECRE-   *
                            MENTS ALONG THE X (HORIZONTAL) AXIS FOR THE  *
                            KTH PLOT. THERE WILL BE NX(K)+1 TIC MARKS    *
                            AND NUMERIC ANNOTATIONS, THE FIRST WILL BE   *
                            FOR X0(K) AND THE LAST FOR XN(K). IF NX(K)   *
                            IS POSITIVE, THE SCALE WILL BE LINEAR, AND   *
                            IF NEGATIVE, THE SCALE WILL BE LOGARITHMIC.  *
                                                                         *
    X0(K)                   THE VALUE OF THE ORIGIN OF THE X AXIS FOR    *
                            THE KTH PLOT.                                *
                                                                         *
    XN(K)                   THE VALUE OF THE END OF THE X AXIS FOR THE   *
                            KTH PLOT. FOR NX(K) POSITIVE, XN(K) SHOULD   *
                            EQUAL X0(K) + NX(K)*DX, WHERE DX IS A REASON- *
                            ABLE PLOT DECREMENT. IF NX(K) IS NEGATIVE,   *
                            BOTH X0(K) AND XN(K) SHOULD BE POWERS OF TEN, *
                            WHERE XN(K) = X0(K)*10**|NX(K)|.             *
                                                                         *
    XL(K)                   THE LENGTH (PLOTTING INCHES) OF THE X AXIS   *
                            FOR THE KTH PLOT. XL(K) SHOULD BE AT LEAST   *
                            ONE INCH LESS THAN XS(K).                    *
                                                                         *
    XS(K)                   THE PAPER SIZE (PLOTTING INCHES) IN THE X    *
                            DIRECTION FOR THE KTH PLOT. THE PLOT WILL BE *
                            CENTERED WITHIN THIS DIMENSION.              *
                                                                         *
CARD I.4.K                  FORMAT (I4, 4X, 4F8.0)                       *
                                                                         *
    NY(K),Y0(K),YN(K),      SAME DEFINITIONS AS FOR THE CORRESPONDING    *
    YL(K) AND YS(K)         ITEMS ON CARD I.3.K BUT FOR THE Y (VERTICAL) *
                            AXIS FOR THE KTH PLOT. NOTE THAT EACH OF THE *
                            NYP(K) VARIABLES WILL BE PLOTTED ON THE SAME *
                            SCALE.                                       *
                                                                         *
NOTE: TO PLOT ON THE VERSATEC PLOTTER AT CALSPAN, THE EXEC CARD SHOULD   *
CONTAIN THE PARAMETERS     ,PLOTTER=VERSATEC,LONG=M                      *
    WHERE M=V INDICATES THAT THE X AXIS WILL BE IN THE LONG (11 INCH)    *
              DIRECTION. FOR THIS CASE, THE RECOMMENDED VALUES FOR       *
              XS(K) AND YS(K) ARE 10.5 AND 8.0.                          *
      AND M=U INDICATES THAT THE Y AXIS WILL BE IN THE LONG DIRECTION,   *
              AND THE RECOMMENDED VALUES FOR XS(K) AND YS(K) ARE REV-    *
              ERSED.                                                     *
                                                                         *
IN ADDITION, THE FOLLOWING CARD IS REQUIRED AT THE END OF THE JOB:       *
    //  EXEC  VPLOT,PCOPY=N                                              *
         WHERE N IS THE NUMBER OF COPIES TO BE PRODUCED.                 *
                                                                         *
```

```
CARD I.5.K                FORMAT (I4, 4X, 15A4)

    NXLAB(K)              THE NUMBER OF CHARACTERS IN THE LABEL OF THE
                          X AXIS FOR THE KTH PLOT (MAX=60, MAY BE ZERO).

    XLAB(K)               THE ALPHANUMERIC INFORMATION TO BE USED AS
                          THE LABEL OF THE X AXIS FOR THE KTH PLOT.
                          DATA SHOULD BE LEFT ADJUSTED AS INPUT SINCE
                          PROGRAM WILL CENTER THE NXLAB(K) CHARACTERS
                          BENEATH THE X AXIS.

CARD I.6.K                FORMAT (I4, 4X, 15A4)

    NYLAB(K),             SAME DEFINITION AS FOR CARD I.5.K BUT FOR THE
    YLAB(K)               LABEL OF THE Y AXIS FOR THE KTH PLOT.

CARD I.7.K                FORMAT (I4, 4X, 15A4)

    NPLB1(K)              THE NUMBER OF CHARACTERS IN THE UPPER OF TWO
                          LINES OF PLOT IDENTIFICATION FOR THE KTH PLOT
                          (MAX = 60, MAY BE ZERO).

    PLB1(K)               THE ALPHANUMERIC INFORMATION TO BE USED IN
                          THE UPPER LINE OF THE PLOT IDENTIFICATION
                          FOR THE KTH PLOT. DATA SHOULD BE LEFT ADJUSTED
                          AS INPUT SINCE THE PROGRAM WILL CENTER THE
                          NPLB1(K) CHARACTERS BENEATH THE X AXIS LABEL.

CARD I.8.K                FORMAT (I4, 4X, 15A4)

    NPLB2(K),             SAME DEFINITION AS FOR CARD I.7.K BUT FOR THE
    PLB2(K)               LOWER LINE OF THE PLOT IDENTIFICATION.

NOTE: THE 15A4 TERM IN THE FORMAT FOR CARDS I.5-I.8 IS TO BE USED
ON COMPUTERS WHERE A SINGLE PRECISION WORD IS EQUIVALENT TO FOUR
ALPHANUMERIC CHARACTERS. THIS TERM IN THE FORMAT FOR SUBROUTINE
POSTPR SHOULD BE TO 10A6 OR 6A10 FOR THOSE COMPUTERS WHOSE SINGLE
PRECISION WORD SIZE IS EQUIVALENT TO 6 OR 10 CHARACTERS. THIS IS
NECESSARY TO INSURE THAT A CONTIGUOUS STRING OF CHARACTERS IS STORED
IN THE COMPUTER MEMORY AS REQUIRED BY SUBROUTINE SYMBOL.
```

APPENDIX B

NUMBERED STOPS WITHIN THE ATB-II MODEL COMPUTER PROGRAM

There are many program stops within the ATB-II model computer program. These are all numbered (in octal to be compatible with most computer systems) and most computer systems will print out the STOP number message (as a condition code on the IBM/360 and IBM/370 systems). Most of the program stops will print out an error message indicating the reason of the program stop. For those produced by the input routines, the actual input error is probably caused by missing or erroneous data on previous input cards. The user is advised to check the output produced by the input routines to ascertain at what point within the input deck the error may have occurred.

Following is a list of all the numbered program stops within the ATB-II model computer program, the subroutine involved, the input card number (where applicable), the reason for the stop and possible remedial action.

1: Main Program; normal program stop, all activity requested by the user input has been completed.

2: Subroutine RSTART, input card A.2; improper variable name, index or type has been supplied.

3: Subroutine BINPUT, input card B.3; error in defining flexible elements, there is only one negative JNT in string.

4: Subroutine BINPUT, input card B.7.A; value of NFX does not agree with the value of NFLX that has been computed from the data supplied on input cards B.3.

5: Subroutine BINPUT, input card B.7.j; the segment number defined by KNT(J) is not an interior segment of a flexible element from data supplied on input cards B.3.

6: Subroutine VINPUT, input card C.2; improper value for MSEG. Allowable values are zero or blank (to represent the primary vehicle), ≤ NSEG (to indicate prescribed motion for one of the specified segments) or one greater than the value of MSEG supplied on a previous C.2 card.

7: Subroutine VINPUT, input cards C; the number of sets of C cards is greater than 6 or the total number of segments defined by the program is greater than 30.

10: Subroutine SINPUT, input card D.2; the plane identification index (J) is in error, must be supplied as consecutive integers.

11: Subroutine KINPUT, input card E.6; the function number is less than 1 or greater than 50.

12: Subroutine KINPUT, input card E.7; the function number is less than 1 or greater than 50.

13: Subroutine KINPUT, input card E.7.D; inconsistent value for THETAO.

14: Subroutine FINPUT, input cards F.1.B-F.4.B; the supplied value for NJ (first number on line just printed) does not correspond to the index J supplied on input cards F.1.A-F.4.B.

15: Subroutine FDINIT, input cards F.1-F.4 (Subroutine FINPUT), F.8.C or F.8.D1 (Subroutine HINPUT); the printed function number has not been defined on input cards E.

16: Subroutine FDINIT, input cards F.1-F.4 (Subroutine FINPUT), F.8.C or F.8.D1 (Subroutine HINPUT); the size of the generated TAB array exceeds 2000 or the size of the NTAB array exceeds 500. These arrays are generated by input cards E and F.

17: Subroutine FINPUT, input cards F.5; the function number has not been defined on input cards E.7.

20: Subroutine FINPUT, input card F.6; the air bag number K has not been supplied in numeric order.

21: Subroutine FINPUT, input card F.7.B; the value of JJ does not correspond to the index J of the non-zero elements read in on input card F.7.A.

24: Subroutine INITAL, input cards G.3; input error for IYPR(4,J), supplied value is greater than J and less than or equal to NSEG.

25: Subroutine INITAL, input cards G.3; input error for IYPR(4,J), supplied value is negative but not equal to -|JNT(J-1)|.

26: Subroutine EQUILB, input card G.4, G.5 or G.6; card number and contents are printed.

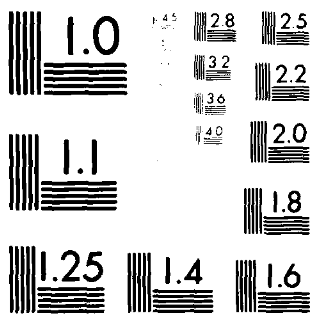
27: Subroutine EQUILB, input cards G.5; iteration for listed variable is not converging within the specified range.

30: Subroutine POSTPR, input card I.9 (on Edgewood Univac 1108 only): card is missing or in error, no plots have been generated.

AD-A088 029 CALSPAN ADVANCED TECHNOLOGY CENTER BUFFALO NY F/G 1/2
ADVANCED RESTRAINT SYSTEM MODELING.(U)
MAY 80 F E BUTLER, J T FLECK F33615-78-C-0516
UNCLASSIFIED CALSPAN-6306-V-1 AFAMRL-TR-80-14 NL

2 OF 2
AD
A088029

END
DATE
FILMED
9-80
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

31: Subroutine DINT; negative square root has been detected in Subroutine PDAUX with the time step size H=HMIN. This is usually an indication that there is extreme angular motion occurring. Unless there are other obvious errors, can be remedied by tightening the angular convergance tests on input cards B.6 or decreasing the value for HMIN on input card A.3.

32: Subroutine AIRBG3; logical error in program code has been detected.

33: Subroutine IMPULS; improper arguments to Subroutine IMPULS, program logic error.

34: Subroutine DAUX; value of NJ2 exceeds the array size for RHS and IJK.

35: Subroutine FSMSOL; maximum dimension of 400 on C array has been exceeded.

36: Function FNTERP; improper arguments to function as indicated by error code as follows:

1 - PHI less than $-\pi$,
2 - PHI greater than π,
3 - THETA less than zero,
4 - THETA greater than π.

37: Subroutine OUTPUT; program logic error, NPRT(4) on input card A.5 is less than or equal to -4 or greater than +4.

40: Subroutine HEDING; program logic error, NPRT(4) on input card A.5 is less than or equal to -4 or greater than +4.

41: Subroutine DSMSOL; matrix supplied to Subroutine DSMSOL (by Subroutine IMPLS2, SEGSEG, EDEPTH or INTERS) is singular.

42: Subroutine HBPLAY; program logic error is determining points that are in play for harness-belt systems.

APPENDIX C

CROSS REFERENCE CHART FOR THE ATB-II MODEL

A computer program has been written to generate a cross reference chart showing the relationships between the ATB-II model subprograms, FORTRAN library routines, CALCOMP plotting routines and the labelled common blocks. The input to the program was obtained from the information produced by the MAP procedure on the Univac 1108 Computer System. It shows the complexity of the 105 subprograms that comprise the ATB-II model and has been used at Calspan to aid in setting up overlay procedures for the CVS-III program.

The chart divides the subprograms into logical blocks, dependent on the program flow, that is very similar to the overlay levels (Strieb, 1976) of the program that is currently being used on the CDC computer at Wright-Patterson Air Force Base. It shows the multiplicity of calls that causes so much difficulty in establishing an overlay procedure for the ATB model on the CDC computers and the relationship of the newly developed subprograms to the rest of the program.

During installation of the program, care should be exercised that the labelled common blocks are loaded into computer storage in the order indicated on the chart since Subroutine POSTPR appends the storage for the unneeded common blocks to COMMON/TEMPVS/ for temporary storage to process output Unit No. 8.

CROSS REFERENCE CHART
AMRL ARTICULATED TOTAL BODY (ATB-II) MODEL
30 OCTOBER 1979

```
          CALLING IMPPIRSIBBVSDSACKFMFIIEDIBACDQTPDIUUAIIHHIDSSDBFSDDDDDDDDFIVFEGEEVHICPSPBBEHHWAABOREIIPHSLLIOHPCVFEPELIDRYCMMDDDXDCI
          ROUTINE IARLISEILIIPSIIIIIIDIHORIIDMZSRDSIPPIHHFSIAEEOHLPAAAAAAAASIINJLFUIEIOLELEELBBIIIGRCDNIOIL IOIUERHERVALTIROPRAAOOODSFI
                  IIITITAIKHHLEHRHHHHIIIUCIHJPPEIAEIDDRPPTEIUTTTHXDUUUUUUUUMISTOOULSRIHEGSLLOPEHRRGTRETICCPHGITDIAHCANTIICTROTTTTTVHAI
                  IHPXIARIDPPITPBPPPPHITIIITUU TGUTIAFBLUUTIXUUTPSAXXXXXXXXSIPEIBHRCRITLSETTHLLDBB HTPEITCLAAIPIHIPDLEIMIV DS3333T SCI
                  I3LVIRCITUUHDUGUUUUIIALJI ST  FXQITDGSLRCI PP3IEHI123334SOIRRHACAOOICPEGRGGATVAG O TRIPSOXXIUNTHOFFLHEIP ES13133 OTI
                  IDTZITHIATTE TITTTTTILBKI TE  S  IEC32SB I I21HGPI2212346LI PTLTDSHIT GFT  V  GG   HSIRITSSITG  SLD E IR G    3 LTI
CALLED            I   I  I            I   I        I       I                I        I                 I     I          I            I
ROUTINE   FREQ    I   I  I            I   I        I       I                I        I                 I     I          I            I
------------------+---I--I------------I---I--------I-------I----------------I--------I-----------------I-----I----------I------------I
MAIN3D      0     I0  I  I            I   I        I       I                I        I                 I     I          I            I
PRIPLT      1     I*1 I  I            I   I        I       I                I        I                 I     I          I            I
PLTXYZ      1     I *1I  I            I   I        I       I                I        I                 I     I          I            I
------------------+---+--I------------I---I--------I-------I----------------I--------I-----------------I-----I----------I------------I
RSTART      1     I*  I1 I            I   I        I       I                I        I                 I     I          I            I
SEARCH      1     I   I*1I            I   I        I       I                I        I                 I     I          I            I
-------------------------+------------I---I--------I-------I----------------I--------I-----------------I-----I----------I------------I
BLKDTA      1     I*  I  I1           I   I        I       I                I        I                 I     I          I            I
BINPUT      1     I*  I  I 1          I   I        I       I                I        I                 I     I          I            I
VINPUT      1     I*  I  I  1         I   I        I       I                I        I                 I     I          I            I
SPLINE      1     I   I  I  *1        I   I        I       I                I        I                 I     I          I            I
DSETD       1     I   I  I  * 1       I   I        I       I                I        I                 I     I          I            I
SINPUT      1     I*  I  I     1      I   I        I       I                I        I                 I     I          I            I
AIRBGI      1     I   I  I     *1     I   I        I       I                I        I                 I     I          I            I
CINPUT      1     I*  I  I       1    I   I        I       I                I        I                 I     I          I            I
KINPUT      1     I   I  I       *1   I   I        I       I                I        I                 I     I          I            I
FINPUT      1     I   I  I       * 1  I   I        I       I                I        I                 I     I          I            I
HINPUT      1     I   I  I       *  1 I   I        I       I                I        I                 I     I          I            I
FDINIT      2     I   I  I         **2I   I        I       I                I        I                 I     I          I            I
--------------------------------------+---I--------I-------I----------------I--------I-----------------I-----I----------I------------I
INITAL      1     I*  I  I            I1  I        I       I                I        I                 I     I          I            I
EQUILB      1     I   I  I            I*1 I        I       I                I        I                 I     I          I            I
BRCIJK      2     I   I  I            I**2I        I       I                I        I                 I     I          I            I
------------------------------------------+--------I-------I----------------I--------I-----------------I-----I----------I------------I
DINT        1     I*  I  I            I   I1       I       I                I        I                 I     I          I            I
ADJUST      1     I   I  I            I   I*1      I       I                I        I                 I     I          I            I
CMPUTE      1     I   I  I            I   I* 1     I       I                I        I                 I     I          I            I
DZP         1     I   I  I            I   I  *1    I       I                I        I                 I     I          I            I
QSET        1     I   I  I            I   I*   1   I       I                I        I                 I     I          I            I
TRIGFS      1     I   I  I            I   I*    1  I       I                I        I                 I     I          I            I
PDAUX       2     I   I  I            I   I* *   2 I       I                I        I                 I     I          I            I
DSETQ       1     I   I  I            I   I      *1I       I                I        I                 I     I          I            I
---------------------------------------------------+-------I----------------I--------I-----------------I-----I----------I------------I
UPDATE      1     I   I  I            I   I*       I1      I                I        I                 I     I          I            I
UPDFDC      1     I   I  I            I   I        I*1     I                I        I                 I     I          I            I
AIRBG3      1     I   I  I            I   I        I* 1    I                I        I                 I     I          I            I
IMPLS2      1     I   I  I            I   I        I*  1   I                I        I                 I     I          I            I
IMPULS      1     I   I  I            I   I        I*   1  I                I        I                 I     I          I            I
HPTURB      1     I   I  I            I   I        I*    1 I                I        I                 I     I          I            I
HSETC       1     I   I  I            I   I        I     *1I                I        I                 I     I          I            I
-----------------------------------------------------------+----------------+--------+-----------------+-----+----------+------------+
```

CROSS REFFERENCE CHART
AMRL ARTICULATED TOTAL BODY (ATB-II) MODEL
30 OCTOBER 1979

```
        CALLING IMPPIRSIBBVSDSACKFNFIIEDIDACDQTPDIUUAIINHIDSSDDFSDDDDDDDDFIVFEGEEVHICPSPBBEHNWAABOREIIPHSLLIOHPCVFEPELIDRYCMMDDDXDCI
        ROUTINE IARLISEILIIPSIIIIIIDINQRIIDMZSRDSIPPIMMPSIAEEOHLPAAAAAAAASIINJLFUIEIOLELEELBBIIIGRCDNIOILIOIUERHERVALTIROPRAAOOODSFI
                IIITITAIKNNLENRNNNNNIIIUCINJPPEIAEIDDRPPTEIUTTTHXDUUUUUUUUMISTOOULSRINEGSLLOPENRRGTRETISCPNGITDIAHCANTIICTROTTTTTVMAI
                INPXIARIDPPITPBPPPPNITIIITUU TGUTIAFBLUUTIXUUTPSAXXXXXXXXSIPEIBNRCRITLSETTNLLDBB HTPEITCLAAIPINIPDLEIMIY DS3333T SCI
                I3LVIRCITUUNDUGUUUUIIALJI ST 'FXQITDGSLRCI PP3IEM11233345OIRRNACAOOICPEGRGGATVAG O TRIPSOXXIUNTHOFFLMEIP ES13133 OTI
                IDTZITHIATTE TITTTTTILBKI TE  S  IEC32SB I 12INGP1221234SLI PTLTDSNIT GFT  Y  GG   HSIRITSSITG  SLD E IR G     3 LTI
CALLED          I   I  I            I   I        I       I                I        I                 I     I          I            I
ROUTINE  FREQ   I   I  I            I   I        I       I                I        I                 I     I          I            I
-------------------------------------------------------------------------------------------------------------------------------------
DAUX       5    I   I  I            I * I      * I*  **  I5               I        I                 I     I          I            I
SETUP1     1    I   I  I            I   I        I       I*1              I        I                 I     I          I            I
SETUP2     2    I   I  I            I   I        I*      I* 2             I        I                 I     I          I            I
BOTT31     1    I   I  I            I   I        I       I  *1            I        I                 I     I          I            I
BNHPIN     1    I   I  I            I   I        I       I  * 1           I        I                 I     I          I            I
FLXSEG     1    I   I  I            I   I        I       I*    1          I        I                 I     I          I            I
SPDAMP     1    I   I  I            I   I        I       I*     1         I        I                 I     I          I            I
DAUX11     1    I   I  I            I   I        I       I*      1        I        I                 I     I          I            I
DAUX12     1    I   I  I            I   I        I       I*       1       I        I                 I     I          I            I
DAUX22     1    I   I  I            I   I        I       I*        1      I        I                 I     I          I            I
DAUX31     1    I   I  I            I   I        I       I*         1     I        I                 I     I          I            I
DAUX32     1    I   I  I            I   I        I       I*          1    I        I                 I     I          I            I
DAUX33     1    I   I  I            I   I        I       I*           1   I        I                 I     I          I            I
DAUX44     1    I   I  I            I   I        I       I*            1  I        I                 I     I          I            I
DAUX55     1    I   I  I            I   I        I       I*             1 I        I                 I     I          I            I
FSMSOL     2    I   I  I            I   I        I     * I*              2I        I                 I     I          I            I
-------------------------------------------------------------------------------------------------------------------------------------
VISPR      2    I   I  I            I   I        I    *  I*               I2       I                 I     I          I            I
FNTERP     1    I   I  I            I   I        I       I                I*1      I                 I     I          I            I
EJOINT     2    I   I  I            I   I        I    *  I*               I  2     I                 I     I          I            I
GLOBAL     2    I   I  I            I   I        I       I                I* *2    I                 I     I          I            I
EFUNCT     2    I   I  I            I   I        I       I                I* * 2   I                 I     I          I            I
EULRAD     1    I   I  I            I   I        I       I                I  *  1  I                 I     I          I            I
VISCOS     2    I   I  I            I   I        I       I                I* *   2 I                 I     I          I            I
HERRON     1    I   I  I            I   I        I       I                I   *   1I                 I     I          I            I
-------------------------------------------------------------------------------------------------------------------------------------
CONTCT     1    I   I  I            I   I        I       I*               I        I1                I     I          I            I
PLELP      2    I   I  I            I   I        I    *  I                I        I*2               I     I          I            I
SEGSEG     2    I   I  I            I   I        I    *  I                I        I* 2              I     I          I            I
PLSEGF     2    I   I  I            I   I        I       I                I        I **2             I     I          I            I
BELTRT     1    I   I  I            I   I        I       I                I        I*   1            I     I          I            I
BELTG      1    I   I  I            I   I        I       I                I        I    *1           I     I          I            I
ELONG      1    I   I  I            I   I        I       I                I        I     *1          I     I          I            I
HBPLAY     2    I   I  I            I*  I        I     * I                I        I       2         I     I          I            I
HBELT      2    I   I  I            I   I        I     * I                I        I*       2        I     I          I            I
W'NDV      1    I   I  I            I   I        I       I                I        I*        1       I     I          I            I
AIRBAG     1    I   I  I            I   I        I       I                I        I*         1      I     I          I            I
AIRBGG     2    I   I  I            I   I        I  *    I                I        I          *2     I     I          I            I
BGG        1    I   I  I            I   I        I       I                I        I           *1    I     I          I            I
ORTHO      1    I   I  I            I   I        I       I                I        I            *1   I     I          I            I
RCRT       1    I   I  I            I   I        I       I                I        I            * 1  I     I          I            I
EDEPTH     1    I   I  I            I   I        I       I                I        I            *  1 I     I          I            I
INTERS     2    I   I  I            I   I        I       I                I        I  *         *   2I     I          I            I
-------------------------------------------------------------------------------------------------------------------------------------
```

CROSS REFERENCE CHART
AMRL ARTICULATED TOTAL BODY (ATB-II) MODEL
30 OCTOBER 1979


```
CALLING   IMPP IRS IBBVSDSACKFHF IIED IDACDOTPD IUUAIIHH IDSSDDF SDDDDDDDDF IVFEGEEVH ICFSFBBEHHVAABORE IIPHSLL IOHPCVFEPEL IDRVCMMDDDXDC I
ROUTINE   IARL ISE IL11PS111111D INOR I IDMZSRDS IPPIHMPS IAEEOHLPAAAAAAAAS IINJLFUIE IOLELEELBBIIIGRCDN IOILIO IUERHERVALT IROPRAAOOODSF I
          I11T ITA IKHHLENRHHHHNI I IUC IHJPPEIAE IDDRPPTE IUTTTHXDUUUUUUUUM ISTOOULSR IHEGSLLOPENRRGTRET ISCPNG ITDIAHCANTI ICTROTTTTTVMA I
          IHPX IAR IDPPITPBPPPPNI TI I ITUU TGUT IAFBLUUT IXUUTPSAXXXXXXXXS IPEIBHRCR ITLSETTHLLDBB HTPE ITCLAA IPINIPDLEIMI IY DS3333T SC I
          I3LY IRC ITUUNDUGUUUUI IALJ I ST  FXQ ITDGSLRC I PP31EMI12333450 IRRNACAQO ICPEGRGGATVAG O TR IPSOXX IUNTHOFFLHE IP ES13133 OT I
          IDTZ ITH IATTE TITTTTT ILBK I TE  S  IEC32SB  I 121NGP12212345L I PTLTDSN IT GFT  V  GG   HS IRITSS ITG  SLD E  IR G      3 LT I
CALLED
ROUTINE   FREQ

POSTPR     1
NICCSI     1
SLPLOT     1
LINAXS     1
LOGAXS     1

OUTPUT     6
HEDING     2
PRINT      6
CHAIN      3
VEMPOS     2
FRCDFL     7
EVALFD     6
PANEL      2
ELTIME    40
LTIME      1

DRCVPR     6
ROT        2
YPRDEG     5
CROSS     23
MAT31     24
MAT33      8
DOT31     29
DOT33      6
DOTT33     7
XDY        7
DSMSOL     4
CFACTT     2

DSQRT     37
DSIN       7
DCOS       6
DASIN      3
DACOS      4
DATAN2     7
DEXP       2
XPDD       4
PLOT       4
SYMBOL     2
NUMBER     1
NSTOPS    22
```

```
                                   CROSS REFERENCE CHART                                      PAGE NO. 4
                        AMRL ARTICULATED TOTAL BODY (ATB-II) MODEL
                                     30 OCTOBER 1979

         CALLING  |MPP|RS|BBVSDSACKFHF|IED|DACDQTPD|UUAIIHH|OSSDDFSDDDDDDDDF|VFEGEEVH|CPSPBBEHHWAABOREI|PHSLL|OHPCVFEPEL|DRVCMMDDDKDC|
         ROUTINE  |ARL|SE|LIIPSIIIIIID|NQR|IDMZSRDS|PPIMMPS|AEEOHLPAAAAAAAAS|INJLFUIE|OLELEELBBIIIGRCDN|OILIO|UERHERVALT|ROPRAAOOODSF|
                  |IIT|TA|KNNLENRNNNNI|IUC|NJPPEIAE|DDRPPTE|UTTTHXDUUUUUUUUM|STOOULSR|NEGSLLOPENRRGTRET|SCPNG|TDIAHCANTI|CTROTTTTTYMA|
                  |NPX|AR|DPPITPBPPPPN|TII|TUU TGUT|AFBLUUT|XUUTPSAXXXXXXXXS|PEIBNRCR|TLSETTNLLDBB HTPE|TCLAA|PINIPDLEIM|Y DS3333T SC|
                  |3LY|RC|TUUNDUGUUUUI|ALJ| ST  FXQ|TDGSLRC| PP3IEM112333450|RRNACAOO|CPEGRGGATVAG O TR|PSOXX|UNTHOFFLME|P ES13133 OT|
                  |DTZ|TH|ATTE TITTTTT|LBK| TE  S  |EC32S8 | IZINGP12212345L| PTLTDSN|T GFT  Y  GG   NS|RITSS|TG  SLD E |R G     3 LT|
COMMON            |   |  |            |   |        |       |                |        |                 |     |          |            |
BLOCK      FREQ   |   |  |            |   |        |       |                |        |                 |     |          |            |
------------------------------------------------------------------------------------------------------------------------------------
CONTRL       51   |***|* | **  ** *** |** |* *   * |* **** |***  ********** |* *     |*   ** * ***   * |*    |*****  *  |            |
CHSNTS       43   |*  |* |*** *** * * |** |** *   *|  *  * |* *  *        * |**** * *|    **   ** *  **|*    |*** *     |***         |
TITLES       16   |** |* | **  **  ** |** |      * |       |                |        |                 |*    |***       |            |
FORCES       15   |   |* |      *     |   |        |* *    |                |* *     |*** *     **     |*    |**        |            |
JBARTZ       13   | * |* |         *  | * |        |* * *  |                |        |*         **     |*    |**        |            |
RSAVE         4   |   |* |            |   |        |       |                |        |                 |*    |**        |            |
CDINT         7   |   |* |            |   |***  *  |       |                |        |                 |**   |          |            |
------------------------------------------------------------------------------------------------------------------------------------
TEMPVS       49   | **|  |***  *******|** |      * |  ** **|***  ********** |* *     | ***** ******    |*    |** *      |            |
------------------------------------------------------------------------------------------------------------------------------------
SGMNTS       43   |***|* |  *  **     |** |      * |* *****|*** **** *  *** |* *     | ****  *****     |     |* ***  *  |            |
DESCRP       31   | * |* |**   *  *   |** |      * |*  **  |*** *  ******** |* **    |          *      |     |* **      |            |
TABLES       24   |   |* |       *****| * |        |**  * *|      *         | * *   *|*****   **       |     |     **   |            |
CHATRX       19   |   |* |            | * |        |*  **  |***   ********* |* *     |                 |     |  *       |            |
CSTRNT       10   |   |* |    *   *   | * |        |*  **  |* *       ***** |        | ***             |     |  *       |            |
CNTSRF       10   |** |* | *   **   * | * |        |  *  * |                |        | ** *  *****     |     |          |            |
FLXBLE        9   |   |* | *          |   |      * |   **  |*    *       ** |        |                 |     |          |            |
CEULER        8   |   |* | *          |   |        |*      |    *    *      |* *     |                 |     |  *       |            |
HRNESS        8   |   |* |          * |   |        |*    **|                |        |*      **        |     |          |            |
TEMPVI        7   |   |* |            |   |        |*   *  |                |* **    |   *             |     |          |            |
COMAIN        6   |*  |* |            |   |***     |       |                |        |                 |     |*         |            |
JNTEST        6   |   |* | **   *     |   |*     * |       |                |        |                 |     |          |            |
VPOSTN        5   |  *|* |  *         |*  |        |       |                |        |                 |     |    *     |            |
ABDATA        5   |   |* |      *     |   |        |  *    |                |        |          **     |     |          |            |
CYDATA        5   |   |* |      *     |   |        |  *    |                |        |          **     |     |          |            |
WINDFR        4   |   |* |         *  |   |        |       |                |        |*        *       |     |          |            |
DAMPER        3   |   |* |     *      |   |        |       |      *         |        |                 |     |          |            |
------------------------------------------------------------------------------------------------------------------------------------
```

APPENDIX D

LIST OF 105 SUBPROGRAMS
THAT COMPRISE THE ATB-II MODEL COMPUTER PROGRAM

This appendix contains a list of the 105 subprograms that comprise the ATB-II model computer program in the order that they were supplied on the program tape ATBMDL sent to WPAFB on 19 November 1979. The first subprogram is merely the common blocks used by the program, the second is the main program followed by all of the remaining subprograms in alphabetical order. Each subprogram name is appended with its revision number followed by the date of the latest change to the subprogram. This same date and revision number appears on the second card of each subprogram.

All subprograms whose revision number are 19 were modified or are new for version 19 of the CVS program developed for this contract. However, only those marked with an asterisk are included in the subprogram listings in Appendix E because they represent major development efforts. The remaining subprograms, marked revision 19, contain only minor changes, primarily a reorganization of the labelled common blocks used by each subprogram, from earlier versions of the program.

LIST OF 105 SUBPROGRAMS

THAT COMPRISE THE ATB-II MODEL COMPUTER PROGRAM

SUBPROGRAM REV. NO.	DATE	SUBPROGRAM REV. NO.	DATE	SUBPROGRAM REV. NO.	DATE
* COMMON 19	10/23/79	DSETD 19	08/05/78	MAT31 17	01/03/77
* MAIN3D 19	10/30/79	DSETQ 19	08/05/78	MAT33 17	01/03/77
ADJUST 19	09/18/79	DSMSOL 03	07/08/74	ORTHO 03	05/31/73
AIRBAG 19	08/05/78	DZP 19	08/05/78	OUTPUT 19	10/05/78
AIRBGG 19	08/05/78	EDEPTH 19	08/05/78	PANEL 19	08/05/78
AIRBG1 19	09/18/79	EFUNCT 10	08/16/74	PDAUX 19	09/05/78
AIRBG3 19	08/05/78	EJOINT 19	10/23/78	PLELP 19	10/19/79
BELTG 19	08/05/78	ELONG 01	10/05/72	PLSEGF 19	10/19/79
BELTRT 19	10/19/79	ELTIME 19	09/18/79	PLTXYZ 19	09/05/78
BGG 19	08/05/78	EQUILB 19	10/19/79	POSTPR 19	02/20/79
BINPUT 19	10/23/78	EULRAD 19	08/05/78	PRINT 19	05/25/79
BLKDTA 19	08/05/78	EVALFD 10	09/26/74	PRIPLT 19	09/05/78
CFACTT 03	05/31/73	* FDINIT 19	06/08/79	QSET 16	03/24/76
CHAIN 19	09/05/78	FINPUT 19	04/27/79	RCRT 03	07/19/73
CINPUT 19	08/05/78	FLXSEG 19	08/05/78	ROT 19	08/05/78
CMPUTE 19	09/18/79	* FNTERP 19	08/05/78	RSTART 19	10/23/79
* CONTCT 19	10/23/79	* FRCDFL 19	10/19/79	SEARCH 19	10/23/79
CROSS 03	05/31/73	FSMSOL 19	04/27/79	SEGSEG 19	10/19/79
DAUX 19	04/27/79	GLOBAL 19	10/19/79	SETUP1 19	08/05/78
DAUX11 19	09/05/78	* HBELT 19	10/23/79	SETUP2 19	08/05/78
DAUX12 19	09/05/78	* HBPLAY 19	10/23/79	SINPUT 19	09/05/78
DAUX22 19	09/05/78	HEDING 19	08/05/78	SLPLOT 18	03/21/78
DAUX31 19	09/05/78	HERRON 19	08/05/78	SPDAMP 19	08/05/78
DAUX32 19	09/05/78	HICCSI 18	07/26/78	* SPLINE 19	05/14/79
DAUX33 19	09/05/78	* HINPUT 19	10/23/79	SPRNGF 19	08/05/78
DAUX44 19	09/05/78	* HPTURB 19	10/23/79	TRIGFS 19	08/05/78
DAUX55 19	09/05/78	* HSETC 19	10/30/79	* UPDATE 19	10/23/79
DHHPIN 19	08/05/78	IMPLS2 19	09/05/78	UPDFDC 19	10/19/79
DINT 19	09/18/79	IMPULS 19	09/05/78	* VEHPOS 19	09/15/78
DOTT31 17	12/20/76	* INITAL 19	05/25/79	* VINPUT 19	06/08/79
DOTT33 17	01/03/77	INTERS 19	08/05/78	VISCOS 19	10/23/78
DOT31 17	01/03/77	* KINPUT 19	09/18/79	* VISPR 19	10/30/79
DOT33 17	01/03/77	LINAXS 18	02/28/78	* WINDY 19	08/05/78
DRCIJK 18	02/24/78	LOGAXS 19	09/18/79	XDY 07	01/31/74
DRCYPR 19	08/05/78	LTIME 01	02/27/74	YPRDEG 19	08/05/78

APPENDIX E

LISTING OF FORTRAN IV SOURCE DECKS OF ATB-II SUBPROGRAMS DEVELOPED FOR WPAFB

This appendix contains a listing of the FORTRAN IV source decks of the ATB subprograms that were developed or substantially modified under Contract Nos. F33615-75-C-5002 and F33615-78-C-0516 for AMRL, Wright-Patterson AFB. Actually, minor modifications were made to many other subprograms (those identified as Revision 19 in Appendix D) but only those subprograms that were new or involve major modifications are included here. They are:

Subprogram Name	Date
COMMON 19	10/23/79
MAIN3D 19	10/30/79
CONTCT 19	10/23/79
FDINIT 19	06/08/79
FNTERP 19	08/05/78
FRCDFL 19	10/19/79
HBELT 19	10/23/79
HBPLAY 19	10/23/79
HINPUT 19	10/23/79
HPTURB 19	10/23/79
HSETC 19	10/30/79
INITAL 19	05/25/79
KINPUT 19	09/18/79
SPLINE 19	05/14/79
UPDATE 19	10/23/79
VEHPOS 19	09/15/78
VINPUT 19	06/08/79
VISPR 19	10/30/79
WINDY 19	08/05/78

```
      BLOCK DATA                                                        COMMON  0010
C                                                   REV 19 10/23/79     COMMON  0020
      IMPLICIT REAL*8 (A-H,O-Z)                                         COMMON  0030
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,           COMMON  0040
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36)        COMMON  0050
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),                         COMMON  0060
     *               UNITL,UNITM,UNITT,GRAVTY(3)                        COMMON  0070
      COMMON/JBARTZ/ MNPL(   30),MNBLT(   8),MNSEG(   30),MNBAG(    6), COMMON  0080
     *               MPL(3,5,30),MBLT(3,5,8),MSEG(3,5,30),MBAG(3,10,6), COMMON  0090
     *               NTPL( 5,30),NTBLT( 5,8),NTSEG( 5,30)               COMMON  0100
      COMMON/TITLES/ DATE(3),COMENT(40),VPSTTL(20),BDYTTL(5),           COMMON  0110
     *               BLTTTL(5,8),PLTTL(5,30),BAGTTL(5,6),SEG(30),       COMMON  0120
     *               JOINT(30),CGS(30),JS(30)                           COMMON  0130
      REAL DATE,COMENT,VPSTTL,BDYTTL,BLTTTL,PLTTL,BAGTTL,SEG,JOINT      COMMON  0140
      LOGICAL*1 CGS,JS                                                  COMMON  0150
      COMMON/FORCES/ PSF(7,30),BSF(4,20),SSF(10,20),BAGSF(3,20),        COMMON  0160
     *               PRJNT(6,30),NPANEL(5),NPSF,NBSF,NSSF,NBGSF         COMMON  0170
      COMMON/RSAVE/  XSG(3,20,3),NSG(7),MSG(20,7)                       COMMON  0180
      COMMON/CDINT/  UU(4),GH(3,4),                                     COMMON  0190
     *               E(3,240),FF(5,240),GG(5,240),Y(5,240),U(5,240),    COMMON  0200
     *               H,HPRINT,TSAVE,TPRINT,TSTART,ICNT,IDBL,IFLAG       COMMON  0210
C     NOTE:  FF REPLACES F.                                             COMMON  0220
      COMMON/TEMPVS/ JTMPVS(10538)                                      COMMON  0230
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30), COMMON  0240
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)       COMMON  0250
      COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(3,60),HA(3,60),HB(3,60), COMMON  0260
     *               RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90),     COMMON  0270
     *               JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30)    COMMON  0280
      COMMON/CNTSRF/ PL(17,30),BELT(20,8),TPTS(6,8),BD(24,40)           COMMON  0290
      COMMON/TABLES/ MXNTI,MXNTB,MXTB1,MXTB2,NTI(50),NTAB(500),TAB(2000) COMMON  0300
      COMMON/VPOSTN/ ZPLT(3),SPLT(3),AXV(3,6),VATAB(6,101,6),           COMMON  0310
     *               VT0(6),VDT(6),TIMEV(6),OMEGV(6),NVTAB(6),INDXV(6)  COMMON  0320
      COMMON/CMATRX/ V1(3,30),V2(3,30),V3(3,12),B12(3,3,60),A22(3,3,60), COMMON  0330
     *               F(3,30),TQ(3,30),WJ(30)                            COMMON  0340
      COMMON/CEULER/ IEULER(30),HIR(3,3,30),ANG(3,30),ANGD(3,30),       COMMON  0350
     *               FE(3,30),TQE(3,30),CONST(3,30)                     COMMON  0360
      COMMON/FLXBLE/ HF(4,12,8),B42(3,3,24),V4(3,8),NFLEX(3,8)          COMMON  0370
      COMMON/CSTRNT/ A13(3,3,24),A23(3,3,24),B31(3,3,24),B32(3,3,24),   COMMON  0380
     *               HHT(3,3,12),RK1(3,12),RK2(3,12),QQ(3,12),TQQ(3,12), COMMON  0390
     *               RQQ(3,12),HQQ(3,12),SQQ(12),CFQQ(12),              COMMON  0400
     *               KQ1(12),KQ2(12),KQTYPE(12)                         COMMON  0410
      COMMON/TEMPVI/ CREST,TTI(3),R1I(3),R2I(3),JSTOP(4,2,30)           COMMON  0420
      COMMON/DAMPER/ APSDM(3,20),APSDN(3,20),ASD(5,20),MSDM(20),MSDN(20) COMMON  0430
      COMMON/INTEST/ SGTEST(3,4,30),XTEST(3,120),SEGT(120),REGT(120)    COMMON  0440
      REAL     SEGT                                                     COMMON  0450
      COMMON/COMAIN/ VAR(240),DER(240),DT,H0,HMAX,HMIN,RSTIME,          COMMON  0460
     *               ISTEP,NSTEPS,NDINT,NEQ,IRSIN,IRSOUT                COMMON  0470
      COMMON/ABDATA/ ZDEP(3,5),DBR(3,3,5),DPVCTR(3,5),DEPLOY(3,5),      COMMON  0480
     *               AB(3,5),B(9,4,5),ZR(3,4,5),BFB(3,4,5),DRR(9,4,5),  COMMON  0490
     *               VBAGG(5),VSCS(5),SPRK(5),CK(5),CMASS(5),CYMIN(5),  COMMON  0500
```

```
*                CYMOUT(5),BAGPV(5),PD(5),VBAG(5),VOLBP(5),
*                PCYV(5),PCYMIN(5),PVBAG(5),TV1(3,4,5),TV2(3,10,5),
*                SWITCH(5),PYMOUT(5),SCALE(5),PREVT,IFULL(6)
 COMMON/CYDATA/  CYTD(5),CYPA(5),CYSP(5),CYTO(5),CYVO(5),CYCD(5),
*                CYK(5),CYR(5),CYAT(5),CYPV(5),CYCDO(5),CYAO(5),
*                CYPO(5),CYSS(5),CYLO(5),CYC(5),CYRHOO(5),CYVMAX(5),
*                CYORFC(5),CYRHO(5),CYT(5),CYP(5),CYV(5)
 COMMON/WINDFR/  WTIME(30),IWIND(30),MWSEG(5,30)
 COMMON/HRNESS/  BAR(15,100),BB(100),BBDOT(100),PLOSS(2,100),
*                XLONG(20),HTIME(2),IBAR(5,100),NL(2,100),
*                NPTSPB(20),NPTPLV(20),NTHRNS(20),NBLTPH(5)
 END
```

```
C     AMRL ARTICULATED TOTAL BODY (ATB-II) MODEL COMPUTER PROGRAM
C     DEVELOPED BY CALSPAN CORP., P.O. BOX 400, BUFFALO, NY 14225
C                                                     REV 19 10/30/79
C     MAIN PROGRAM
C
C     PERFORMS CARD INPUT, PROGRAM INITIALIZATION,
C     CONTROL OF INTEGRATION LOOP AND SELECTED OUTPUT.
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36)
      COMMON/CNTSRF/ PL(17,30),BELT(20,8),TPTS(6,8),BD(24,40)
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)
      COMMON/TITLES/ DATE(3),COMENT(40),VPSTTL(20),BDYTTL(5),
     *               BLTTTL(5,8),PLTTL(5,30),BAGTTL(5,6),SEG(30),
     *               JOINT(30),CGS(30),JS(30)
      REAL DATE,COMENT,VPSTTL,BDYTTL,BLTTTL,PLTTL,BAGTTL,SEG,JOINT
      LOGICAL*1 CGS,JS
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),
     *               UNITL,UNITM,UNITT,GRAVTY(3)
      COMMON/COMAIN/ VAR(240),DER(240),DT,HO,HMAX,HMIN,RSTIME,
     *               ISTEP,NSTEPS,NDINT,NEQ,IRSIN,IRSOUT
      LOGICAL NPRT1,NPRT2,NPRT3
      CALL ELTIME(1, 1)
      CALL BLKDTA
C
C     INPUT CARDS A.1 AND A.2, TEST FOR RESTART.
C
      READ  (5,10) DATE,IRSIN,IRSOUT,RSTIME,COMENT
   10 FORMAT(3A4,2I4,F8.0/20A4/20A4)
      WRITE (6,11) DATE,IRSIN,IRSOUT,RSTIME,COMENT
   11 FORMAT('1',30X,'AMRL ARTICULATED TOTAL BODY (ATB-II) MODEL'   ////
     * 31X,'DEVELOPED BY CALSPAN CORP., P.O. BOX 400, BUFFALO NY 14225'/
     */31X,'FOR THE AEROSPACE MEDICAL RESEARCH LABORATORY,'             /
     * 31X,'AFSC AERONAUTICAL SYSTEMS DIVISION, WRIGHT PATTERSON AFB'   /
     * 31X,'UNDER CONTRACTS F33615-75C-5002 AND F33615-78C-0516.'      //
     * 31X,'AND FOR THE NATIONAL HIGHWAY TRAFFIC SAFETY ADMINISTRATION,'
     */31X,'U.S. DEPARTMENT OF TRANSPORTATION, UNDER CONTRACTS'         /
     * 31X,'FH-11-7592, HS-053-2-485, HS-5-01300 AND HS-6-01410.'    ////
     * 31X,'PROGRAM DOCUMENTATION - NHTSA REPORT NOS. DOT-HS-801-507'   /
     * 31X,'THROUGH 510 (FORMERLY CALSPAN REPORT NO. ZQ-5180-L-1)'      /
     * 31X,'AVAILABLE FROM NTIS (ACCESSION NOS. PB-241692,3,4 AND 5)'   /
     * 31X,'AND APPENDIXES A-I TO THE ABOVE, AVAILABLE FROM CALSPAN;'   /
     * 31X,'AND REPORT NO. AMRL-TR-75-14, AVAILABLE FROM NTIS.'      ////
     * 31X,'PROGRAM ATB-II, EXECUTED ON THE CYBER COMPUTER SYSTEM,'     /
     * 31X,'AFSC AERONAUTICAL SYSTEMS DIVISION, WRIGHT PATTERSON AFB.'//
     *    // 4X,3A4,'  IRSIN=',I4,'  IRSOUT=',I4,'  RSTIME =',F8.4,
     *       61X,'CARDS A'//1X,20A4/1X,20A4//)
      IF (IRSIN.NE.0) GO TO 18
```

```
C
C     INPUT CARDS A.3,A.4 AND A.5.
C
      READ  (5,12) UNITL,UNITM,UNITT,GRAVTY,G
   12 FORMAT(3A4,4F12.0)
      IF  (G.EQ.0.0)  G = DSQRT(GRAVTY(1)**2+GRAVTY(2)**2+GRAVTY(3)**2)
      READ  (5,13) NDINT,NSTEPS,DT,HO,HMAX,HMIN,NPRT
   13 FORMAT(2I4,4F8.0/36I2)
      WRITE (6,14) UNITL,UNITM,UNITT,GRAVTY,
     *             NDINT,NSTEPS,DT,HO,HMAX,HMIN
   14 FORMAT(5X,'UNITL = ',A4,5X,'UNITM = ',A4,5X,'UNITT = ',A4,
     *       5X,'GRAVITY VECTOR = (',F9.4,',',F9.4,',',F9.4,')'//
     *       5X,'NDINT =',I4,5X,'NSTEPS =',I5,5X,'DT =',F8.6,
     *       5X,'HO =',F8.6,5X,'HMAX =',F8.6,5X,'HMIN =',F8.6)
      WRITE (6,15) (I,I=1,36),NPRT
   15 FORMAT('0 NPRT ARRAY'/3X,36I3/3X,36I3)
      NPRT4 = NPRT(4)
      IF  (NPRT(4).LT.0)  GO TO 50
C
C     CALL INPUT ROUTINES
C
      CALL BINPUT
      CALL VINPUT
      CALL SINPUT
      CALL CINPUT
C
C     PROGRAM INITIALIZATION
C
      TIME = 0.0
      CALL INITAL
      GO TO 19
C
C     READ INPUT DATA FROM RESTART TAPE AND WRITE NEW TAPE.
C     THE FIVE FUNCTIONS OF SUBROUTINE RSTART ARE:
C        1. READ INPUT & INITIALIZATION RECORD FROM OLD RESTART TAPE.
C        2. WRITE INPUT & INITIALIZATION RECORD ONTO NEW RESTART TAPE.
C        3. READ TIME POINT RECORD FROM OLD RESTART TAPE.
C        4. READ NEW INPUT DATA FROM INPUT STREAM FOR RESTART.
C        5. WRITE TIME POINT RECORD ONTO NEW RESTART TAPE.
C
   18 CALL RSTART(1,IRSIN)
      CALL RSTART(4,5)
      NPRT4 = NPRT(4)
   19 IF (IRSOUT.NE.0) CALL RSTART(2,IRSOUT)
C
C     INTEGRATION LOOP - ADVANCE TIME BY EITHER INTEGRATING BY
C     SUBROUTINE DINT OR BY FETCHING TIME POINT RECORD FROM RESTART TAPE
C
      TIME = 0.0
      ISTEP = 0
```

```
   20 IF (IRSIN.EQ.0) GO TO 23
      IF (TIME.GT.RSTIME+0.5*DT) GO TO 23
      IF (DABS(TIME-RSTIME).LT.0.5*DT) GO TO 21
      CALL RSTART(3,IRSIN)
      GO TO 24
   21 CALL RSTART(4,5)
      IF  (NPRT(4).LT.0)  GO TO 50
   23 CALL DINT
   24 CALL PRIPLT
      IF (IRSOUT.NE.0) CALL RSTART(5,IRSOUT)
C
C     OPTIONAL OUTPUT
C
      NPRT3 = (NPRT(3).EQ.1)
      IF (NPRT(3).GT.1) NPRT3 = (MOD(ISTEP,NPRT(3)).EQ.0)
      IF (NPRT3) CALL PRINT(6HMAIN3D)
C
C     TAPE 1 OUTPUT - IDENTIFICATION RECORD
C
      NBG1 = NVEH + 1
      NBG2 = NVEH + NBAG
      IF (ISTEP.EQ.0.AND.NPRT(1).NE.0)
     *           WRITE (1) NSEG,NJNT,NBLT,NBAG,NPL,
     *                     DATE,COMENT,VPSTTL,BDYTTL,
     *                     ((BLTTTL(I,J),I=1,5),J=1,NBLT),
     *                     (( PLTTL(I,J),I=1,5),J=1,NPL  ),
     *                     ((BAGTTL(I,J),I=1,5),J=1,NBAG),
     *                     (SEG(J),J=1,NSEG),
     *                     (JOINT(J),J=1,NJNT),
     *                     (CGS(J),J=1,NSEG),
     *                     (JS(J),J=1,NJNT),
     *                     ((   BD(I,J),I=1,3),J=1,NSEG),
     *                     ((   BD(I,J),I=4,6),J=1,NSEG),
     *                     ((     BD(I,J),I=1,3),J=NBG1,NBG2),
     *                     ((BELT(I,J),I=1,6),J=1,NBLT),
     *                     ((PL(I,J),I=1,17),J=1,NPL)
       NPRT1 = (NPRT(1).EQ.1)
      IF (NPRT(1).GT.1) NPRT1 = (MOD(ISTEP,NPRT(1)).EQ.0)
C
C     TAPE 1 OUTPUT - TIME POINT RECORD
C
      IF (NPRT1) WRITE (1) TIME,(SEGLP(I,NVEH),I=1,3),
     *                     ((D(I,J,NVEH),I=1,3),J=1,3),
     *                     ((SEGLP(I,J),I=1,3),J=1,NSEG),
     *                     ((SEGLP(I,J),I=1,3),J=NBG1,NBG2),
     *                     ((( D(K,I,J),K=1,3),I=1,3),J=1,NSEG),
     *                     ((( D(K,I,J),K=1,3),I=1,3),J=NBG1,NBG2),
     *                     ((     BD(I,J),I=4,6),J=NBG1,NBG2)
     *                     ((TPTS(I,J),I=1,6),J=1,NBLT)
       NPRT2 = (NPRT(2).EQ.1)
```

```
      IF (NPRT(2).GT.1) NPRT2 = (MOD(ISTEP,NPRT(2)).EQ.0)
      IF (NPRT2) CALL ELTIME(2,1)
      ISTEP = ISTEP+1
      IF (ISTEP.LE.NSTEPS) GO TO 20
   50 IF  (NPRT4.GT.0)  END FILE 8
      IF  (NPRT(4).EQ.0 .OR. NPRT(4).EQ.4)  GO TO 60
      PRDT = 1000.0*DT
      CALL POSTPR (PRDT)
      IF  (NPRT2)  CALL ELTIME (2,1)
   60 IF  (.NOT.NPRT2)  CALL ELTIME (2,1)
      STOP  1
      END
```

```
      SUBROUTINE CONTCT
C                                                   REV 19 10/23/79
C     CONTROLS THE CALLING OF SUBROUTINES REQUIRED TO COMPUTE THOSE
C     EXTERNAL FORCES AND TORQUES ACTING ON THE BODY SEGMENTS.
C
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36)
      COMMON/JBARTZ/ MNPL(    30),MNBLT(    8),MNSEG(    30),MNBAG(     6),
     *               MPL(3,5,30),MBLT(3,5,8),MSEG(3,5,30),MBAG(3,10,6),
     *               NTPL( 5,30),NTBLT( 5,8),NTSEG( 5,30)
      COMMON/FORCES/ PSF(7,30),BSF(4,20),SSF(10,20),BAGSF(3,20),
     *               PRJNT(6,30),NPANEL(5),NPSF,NBSF,NSSF,NBGSF
      COMMON/TABLES/MXNTI,MXNTB,MXTB1,MXTB2,NTI(50),NTAB(500),TAB(2000)
      COMMON/HRNESS/ BAR(15,100),BB(100),BBDOT(100),PLOSS(2,100),
     *               XLONG(20),HTIME(2),IBAR(5,100),NL(2,100),
     *               NPTSPB(20),NPTPLY(20),NTHRNS(20),NBLTPH(5)
      COMMON/WINDFR/ WTIME(30),IWIND(30),MWSEG(5,30)
      CALL ELTIME(1,12)
      NPSF = 0
      NBSF = 0
      NSSF = 0
      IF (NPL.LE.0) GO TO 21
C
C     CALL PLELP ROUTINE FOR EACH ALLOWED PLANE-SEGMENT CONTACT.
C
      DO 20 J=1,NPL
      IF(MNPL(J).EQ.0) GO TO 20
      KPL = MNPL(J)
      DO 19 I=1,KPL
      NPSF = NPSF+1
      M1 = MPL(1,I,J)
      M2 = MPL(2,I,J)
      M3 = MPL(3,I,J)
      NT = NTPL(I,J)
      JT = NTAB(NT)
      TAB(JT) = 0.0
   19 CALL PLELP(M2,M3,M1,J,NT)
   20 CONTINUE
   21 IF(NBLT.LE.0) GO TO 41
C
C     CALL BELTRT ROUTINE FOR EACH ALLOWED BELT-SEGMENT CONTACT.
C
      DO 30 J=1,NBLT
      IF(MNBLT(J).EQ.0) GO TO 30
      KBLT = MNBLT(J)
      DO 29 I=1,KBLT
      NBSF = NBSF+1
      M1 = MBLT(1,I,J)
      M2 = MBLT(2,I,J)
```

```
      M3 = MBLT(3,I,J)
      NT = NTBLT(I,J)
      JT = NTAB(NT)
      TAB(JT) = 0.0
      NF = NTAB(NT+5)
      IF (NF.NE.0) JT = NTAB(NT+6)
      IF (NF.NE.0) TAB(JT) = 0.0
   29 CALL BELTRT(M2,M3,M1,J,NT)
   30 CONTINUE
C
C     CALL SEGSEG ROUTINE FOR EACH ALLOWED SEGMENT-SEGMENT CONTACT.
C
   41 DO 50 J=1,NSEG
      IF(MNSEG(J).EQ.0) GO TO 50
      KSEG = MNSEG(J)
      DO 49 I=1,KSEG
      NSSF = NSSF+1
      M1 = MSEG(1,I,J)
      M2 = MSEG(2,I,J)
      M3 = MSEG(3,I,J)
      NT = NTSEG(I,J)
      JT = NTAB(NT)
      TAB(JT) = 0.0
   49 CALL SEGSEG(J,M1,M2,M3,NT)
   50 CONTINUE
C
C     CALL AIRBAG ROUTINE FOR ALLOWED BAG-SEGMENT CONTACTS, IF ANY.
C
      IF (NBAG.NE.0) CALL AIRBAG
C
C     CALL WINDY ROUTINE FOR WIND FORCES ON EACH SEGMENT.
C
      DO 60 J=1,NSEG
      IF (MWSEG(1,J).EQ.0) GO TO 60
      M1 = MWSEG(2,J)
      M2 = MWSEG(3,J)
      M3 = MWSEG(4,J)
      NT = MWSEG(5,J)
      CALL WINDY (J,M1,M2,M3,NT)
   60 CONTINUE
C
C     CALL HBELT ROUTINE FOR EACH HARNESS-BELT SYSTEM.
C
      IF (NHRNSS.LE.0) GO TO 99
      J1 = 1
      KNLO = 0
      DO  70  I=1,NHRNSS
      IF  (NBLTPH(I).LE.0)  GO TO 70
      J2 = J1 + NBLTPH(I) - 1
      CALL HBELT (J1,J2,KNLO,0)
```

```
      J1 = J2+1                                                         CONTCT
   70 CONTINUE                                                          CONTCT
   99 CALL ELTIME(2,12)                                                 CONTCT
      RETURN                                                            CONTCT
      END                                                               CONTCT
```

```
      SUBROUTINE FDINIT
C                                                    REV 19 06/08/79
C     REPLACES CODE PREVIOUSLY IN SUBROUTINES FINPUT AND HINPUT.
C     FROM FIVE FUNCTION NUMBERS IN NF ARRAY
C      1. SET UP KTITLE
C      2. SET UP NTAB AND TAB ARRAYS
C      3. INCREMENT COUNTERS MXNTB AND MXTB2
C
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/TABLES/ MXNTI,MXNTB,MXTB1,MXTB2,NTI(50),NTAB(500),TAB(2000)
      COMMON/TEMPVS/ JTITLE(5,51),NF(5),MS(3),KTITLE(31)
C     NOTE: THIS IS SHARED BY SUBS CINPUT, FINPUT, HINPUT AND FDINIT.
      REAL JTITLE,KTITLE
      J1 = MXTB2 + 1
      NT = MXNTB + 1
      NTAB(NT) = J1
      NT = NT+1
      DO 56 L=1,5
      NX = IABS(NF(L))
      NTAB(NT) = 0
      IF (NX.EQ.0) GO TO 56
      NTAB(NT) = ISIGN(NTI(NX),NF(L))
      DO 51 KK = 1,5
      KJ = 5*L+KK+1
   51 KTITLE(KJ) = JTITLE(KK,NX)
      IF (NTI(NX).NE.0) GO TO 56
      WRITE(6,54) NX
   54 FORMAT ('0 FUNCTION NO.',I4,' HAS NOT BEEN DEFINED. ',
     *        '  PROGRAM TERMINATED.')
      STOP 15
   56 NT = NT+1
C
C     INITIALIZE TAB ARRAY TO ZERO EXCEPT FOR DMAX, DINER, FDMAX.
C
      J2 = J1+29
      DO 57 JJ=J1,J2
   57 TAB(JJ) = 0.0
      NX = NTAB(NT-5)
      IF (NX.LT.0) GO TO 58
      TAB(J1+8) = DABS(TAB(NX+1))
      IF (TAB(NX+2).NE.0.0) TAB(J1+8) = DABS(TAB(NX+2))
      DX = TAB(J1+8)
      TAB(J1+10) = EVALFD(DX,NX,1)
      NX = NTAB(NT-4)
      IF (NX.LE.0) GO TO 58
      TAB(J1+9) = DABS(TAB(NX+1))
      IF (TAB(NX+2).NE.0.0) TAB(J1+9) = DABS(TAB(NX+2))
   58 J1 = J2+1
      MXNTB = NT-1
      MXTB2 = J1-1
```

```
      IF (MXTB2.GT.2000) WRITE (6,62) MXTB2                            FDINIT  0510
   62 FORMAT ('0 ERROR IN SUBROUTINE FDINIT, SIZE OF TAB ARRAY =',I8//  FDINIT  0520
     *        '     PROGRAM TERMINATED.')                              FDINIT  0530
      IF (MXNTB.GT.500)  WRITE (6,63) MXNTB                            FDINIT  0540
   63 FORMAT ('0 ERROR IN SUBROUTINE FDINIT, SIZE OF NTAB ARRAY =',I8// FDINIT  0550
     *        '     PROGRAM TERMINATED.')                              FDINIT  0560
      IF  (MXTB2.GT.2000 .OR. MXNTB.GT.500)  STOP 16                   FDINIT  0570
      RETURN                                                           FDINIT  0580
      END                                                              FDINIT  0590
```

```
      DOUBLE PRECISION FUNCTION FNTERP(THETA,PHI,NT)
C                                                     REV 19 08/05/78
C     COMPUTES THE RESTORING TORQUE OF A JOINT AS A FUNCTION OF THE
C     FLEXURE ANGLE (THETA) AND THE AZIMUTH ANGLE (PHI) AS DEFINED BY
C     FUNCTION NO. NT
C
C     ASSUMES  0  < THETA < PI
C             -PI <  PHI  < PI
C              DATA IN TAB ARRAY CONTAINS NTHETA,NPHI FOLLOWED BY
C              TWO DIMENSIONAL ARRAY OF FUNCTIONAL VALUES (NTHETA > 0)
C              OR POLYNOMIAL COEFFICIENTS (NTHETA < 0) FOR EQUALLY
C              SPACED VALUES OF PHI.
C
C                   THETA(I) = (I-1)*PI/(NTHETA-1) FOR I=1,NTHETA
C                   PHI(J) = -PI + (J-1)*2*PI/NPHI FOR J=1,NPHI
C                       F(THETA,PI) = F(THETA,-PI)
C
C     SUBROUTINE EVALUATES G1(THETA) = F(THETA,PHI(J)  )
C                          G2(THETA) = F(THETA,PHI(J+1))
C                FOR  PHI(J) < PHI < PHI(J+1)
C     BY LINEAR INTERPOLATION OR POLYNOMIAL EVALUATION AND THEN LINEAR
C     INTERPOLATES BETWEEN G1 AND G2 TO OBTAIN F(THETA,PHI).
C     IF F < 0, F IS SET TO ZERO, THEREFORE A DEAD BAND IS OBTAINED
C     BY NEGATIVE VALUES IN THE TABLE.
C
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),
     *               UNITL,UNITM,UNITT,GRAVTY(3)
      COMMON/TABLES/MXNTI,MXNTB,MXTB1,MXTB2,NTI(50),NTAB(500),TAB(2000)
      IERROR = 0
      IF (PHI.LT.-PI)    IERROR = 1
      IF (PHI.GT. PI)    IERROR = 2
      IF (THETA.LT.0.0) IERROR = 3
      IF (THETA.GT.PI  ) IERROR = 4
      IF (IERROR.NE.0)  WRITE (6,11) IERROR,THETA,PHI,NT
   11 FORMAT('0 IMPROPER ARGUMENTS TO FUNCTION FNTERP. ERROR CODE =',I4/
     *       '0 THETA =',G25.15, ' PHI =',G25.15,' NT =',I6)
      IF (IERROR.NE.0) STOP 36
      NF = NTI(NT) + 5
      NTHETA = TAB(NF)
      NPHI   = TAB(NF+1)
C
C     DETERMINE INDEX AND INTERPOLATION PARAMETERS FOR PHI.
C
      XNP = (PHI+PI)/(2.0*PI)*TAB(NF+1)
      NP1 = XNP
      NP2 = NP1+1
      IF (NP2.GE.NPHI) NP2 = 0
      RP2 = XNP - DFLOAT(NP1)
      RP1 = 1.0 - RP2
```

```
      NTH = IABS(NTHETA)                                                FNTERP  0510
      IP1 = NF+1+NP1*NTH                                                FNTERP  0520
      IP2 = NF+1+NP2*NTH                                                FNTERP  0530
C                                                                       FNTERP  0540
C     DETERMINE INDEX AND INTERPOLATION PARAMETERS FOR THETA.           FNTERP  0550
C                                                                       FNTERP  0560
      IF (NTHETA.LT.0) GO TO 20                                         FNTERP  0570
      XNT = THETA/PI*(TAB(NF)-1.0)                                      FNTERP  0580
      NT1 = XNT                                                         FNTERP  0590
      RT2 = XNT - DFLOAT(NT1)                                           FNTERP  0600
      RT1 = 1.0 - RT2                                                   FNTERP  0610
      IT1 = IP1 + NT1                                                   FNTERP  0620
      IT2 = IP2 + NT1                                                   FNTERP  0630
      G1  = RT1*TAB(IT1+1) + RT2*TAB(IT1+2)                             FNTERP  0640
      G2  = RT1*TAB(IT2+1) + RT2*TAB(IT2+2)                             FNTERP  0650
      GO TO 23                                                          FNTERP  0660
C                                                                       FNTERP  0670
C     COMPUTE FOR POLYNOMIALS IN THETA FOR FIXED PHI.                   FNTERP  0680
C                                                                       FNTERP  0690
   20 NPOLY = -NTHETA-1                                                 FNTERP  0700
      IT1 = IP1 + NPOLY + 2                                             FNTERP  0710
      IT2 = IP2 + NPOLY + 2                                             FNTERP  0720
      THETA1 = THETA - TAB(IP1+1)                                       FNTERP  0730
      THETA2 = THETA - TAB(IP2+1)                                       FNTERP  0740
      G1 = 0.0                                                          FNTERP  0750
      G2 = 0.0                                                          FNTERP  0760
      DO 21 I=1,NPOLY                                                   FNTERP  0770
      IT1 = IT1-1                                                       FNTERP  0780
      IT2 = IT2-1                                                       FNTERP  0790
      G1 = THETA1*(TAB(IT1)+G1)                                         FNTERP  0800
   21 G2 = THETA2*(TAB(IT2)+G2)                                         FNTERP  0810
   23 FNTERP = RP1*G1 + RP2*G2                                          FNTERP  0820
      IF (FNTERP.LT.0.0) FNTERP = 0.0                                   FNTERP  0830
      RETURN                                                            FNTERP  0840
      END                                                               FNTERP  0850
```

```
      SUBROUTINE FRCDFL (D,RATE,M,N,FRCDF,ELOSS)
C                                                    REV 19 10/19/79
C     EVALUATE FORCE DEFLECTION FUNCTION AT POINT D, WHERE DEFINITION
C     OF FUNCTION IS CONTROLLED BY M INDEX OF NTAB ARRAY.
C     DERIVATIVE, FUNCTION OR INTEGRAL IS EVALUATED AS N = 0,1 OR 2.
C        NTAB(M)     -  INDEX TO TAB ARRAY FOR REAL DATA
C        NTAB(M+1)   -  INDEX TO TAB ARRAY FOR BASE FUNCTION
C        NTAB(M+2)   -  INDEX TO TAB ARRAY FOR INERTIAL FUNCTION, IF ANY
C
C     ASSUMES  0 < DG < DCUBIC < DREF < DMAX
C                BUT ANY < MAY BE LESS THAN OR EQUAL TO
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/TABLES/MXNTI,MXNTB,MXTB1,MXTB2,NTI(50),NTAB(500),TAB(2000)
      F = 0.0
      ELOSS = 0.0
      L = NTAB(M)
      TAB(L) = D
      IF (D.LT.0.0) GO TO 99
      DMAX = TAB(L+8)
      IF (D.LT.DMAX) GO TO 10
C
C     DMAX < D , USE MAX VALUE
C
      IF (N-1)  99,9,99
    9 FDMAX = TAB(L+10)
      F = FDMAX
      GO TO 40
   10 DREF = TAB(L+7)
      IF (D.GE.DREF) GO TO 30
      DCUBIC = TAB(L+6)
      IF (DCUBIC.GE.DREF) GO TO 20
      IF (D.LE.DCUBIC) GO TO 20
C
C     DCUBIC < D < DREF , USE CUBIC
C
      LC = L+14
      DC0 = TAB(L+18)
      X = D-DC0
      IF (N-1) 12,11,99
C
C     USE CUBIC DEFINITION
C
   11 F = TAB(LC) + X *(TAB(LC+1)+X*(TAB(LC+2)+X*TAB(LC+3)))
      GO TO 40
C
C     USE DERIVATIVE OF CUBIC
C
   12 F = TAB(LC+1)+X*(2.0*TAB(LC+2)+X*3.0*TAB(LC+3))
      GO TO 99
```

```
   20 DG = TAB(L+5)
      IF (D.LE.DG) GO TO 40
C
C     DG < D < DCUBIC , USE QUADRATIC
C
      LQ = L+11
      X = D-DG
      IF (N-1) 22,21,99
C
C     USE QUADRATIC DEFINITION
C
  21  F = TAB(LQ)+X*(TAB(LQ+1)+X*TAB(LQ+2))
      GO TO 40
C
C     USE DERIVATIVE OF QUADRATIC.
C
  22  F = TAB(LQ+1)+X*2.0*TAB(LQ+2)
      GO TO 99
C
C     DREF < D < DMAX, USE BASE FUNCTION
C
  30  IF (N-1) 31,31,99
  31  NB = NTAB(M+1)
C
C     EVALUATE BASE FUNCTION
C
      IF (NB.GT.0) F = EVALFD(D,NB,N)
      NI = NTAB(M+2)
C
C     ADD INERTIAL FUNCTION , IF ANY
C
      IF (NI.GT.0) F = F+EVALFD(D,NI,N)
   40 IF  (N.NE.1)  GO TO 99
C
C     COMPUTE AND ADD RATE DEPENDENT FUNCTIONS, IF ANY.
C
C     CURRENT RESTRICTIONS:
C
C        1) COMPUTED FOR N=1 (FUNCTION) ONLY.
C
C        2) FUNCTION NOS. M+2,M+3 AND M+4 (USED FOR INERTIAL SPIKE,
C           R FACTOR AND G FACTOR FUNCTIONS) MUST BE NEGATIVE OR ZERO,
C           I.E., THESE FUNCTIONS CANNOT BE USED IN CONJUNCTION WITH
C           THE RATE DEPENDENT FUNCTIONS.
C
C        3) ASSUMES THE FUNCTIONAL FORM
C
C                 F(D,D') = F1(D) + F2(D)*F3(D') + F4(D')
C
C           WHERE F1(D ) IS DEFINED BY FUNCTION NTAB(M+1)>0.
```

```
C                             I.E., NORMAL FORCE DEFLECTION FUNCTION WITH NO
C                             INERTIAL SPIKE FUNCTION AND DEFAULT VALUES
C                             R=1 AND G=0 (UNLOADING AND RELOADING SAME AS
C                             ORIGINAL LOADING):
C
C                      F2(D ) IS DEFINED BY FUNCTION NTAB(M+2)<0,
C                             IF NTAB(M+2)=0, F2(D )=0;
C
C                      F3(D') IS DEFINED BY FUNCTION NTAB(M+3)<0,
C                             IF NTAB(M+3)=0, F3(D')=0;
C
C                AND   F4(D') IS DEFINED BY FUNCTION NTAB(M+4)<0,
C                             IF NTAB(M+4)=0, F4(D')=0.
C
C                NOTE: FUNCTIONAL FORM CAN BE CHANGED BY REVISING PROGRAM
C                BETWEEN STATEMENTS 40 AND 99.
C
      F2 = 0.0
      F3 = 0.0
      F4 = 0.0
      N2 = -NTAB(M+2)
      N3 = -NTAB(M+3)
      N4 = -NTAB(M+4)
      IF  (N2.GT.0)  F2 = EVALFD (D,    N2,N)
      IF  (N3.GT.0)  F3 = EVALFD (RATE,N3,N)
      IF  (N4.GT.0)  F4 = EVALFD (RATE,N4,N)
      F = F + F2*F3 + F4
      ELOSS = RATE*(F2*F3+F4)
   99 FRCDF = F
      RETURN
      END
```

```
      SUBROUTINE HBELT (J1,J2,KNLO,IND)
C                                                      REV 19 10/23/79
C     ARGUMENTS:
C         J1,J2 - FIRST AND LAST INDEX FOR BELTS.
C          KNLO - ZERO VALUE FOR KNL INDEX.
C          IND  - 0: CALL IS FROM SUBROUTINE CONTCT
C                 1: CALL IS FROM SUBROUTINE UPDATE
C
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/CNTSRF/ PL(17,30),BELT(20,8),TPTS(6,8),BD(24,40)
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)
      COMMON/TABLES/ MXNTI,MXNTB,MXTB1,MXTB2,NTI(50),NTAB(500),TAB(2000)
      COMMON/HRNESS/ BAR(1[illegible],100),BB(100),BBDOT(100),PLOSS(2,100),
     *               XLONG([illegible]),HTIME(2),IBAR(5,100),NL(2,100),
     *               NPTSPB(20),NPTPLY(20),NTHRNS(20),NBLTPH(5)
C     THIS COMMON/TEMPVS/ IS SHARED BY HPTURB, HBPLAY, HBELT AND HSETC.
      COMMON/TEMPVS/ B(3,3,3),S(3,3),T(3),R(3),V(3),T1(3),T2(3),
     *               E(3,3,50),EDOT(3,50),FCE(3,50),FR(3,50),ZR(3,50),
     *               TR(3,50),U(3,50),PTLOSS(2,50),BL(50),FB(50),FP(50),
     *               OLDBB(100),RHS(3,54),C(3,3,200),IJK(54,54)
      CALL ELTIME (1,38)
      NTP = 0
      K2 = 0
      DO  31  JB=J1,J2
      IF  (NPTPLY(JB).LE.0)  GO TO 31
C
C     FIRST LOOP ON K
C     COMPUTE Z(K),ZR(K),E3(K),U(K-1),BL(K-1),FB(K-1)
C     NEED    NL(K),BB(K-1)
C     NOTE: AN INDEX K-1 REFERS TO BELT SEGMENT BETWEEN K-1 AND K.
C
      K1 = K2 + 1
      K2 = K2 + NPTPLY(JB)
      DO  20  K=K1,K2
      KNL = KNLO + K
      KI = NL(1,KNL)
C
C     HERE K   IS INDEX OF POINTS IN PLAY ON EACH HARNESS
C          KNL IS INDEX OF ALL POINTS IN PLAY
C          KI  IS INDEX OF ALL POINTS
C
      KS   = IABS(IBAR(1,KI))
      IF  (KS.GT.100)  NTP = 1
      IF  (KS.GT.100)  KS = MOD(KS,100)
      KE = IBAR(2,KI)
      CALL DOT31 (D(1,1,KS),BAR(4,KI),T1)
      CALL DOT31 (D(1,1,KS),BAR(7,KI),T2)
      DO  11  J=1,3
      R(J) = V(J)
```

```
      V(J) = BAR(J+3,KI) + BAR(J+6,KI)
      TR(J,K) = T1(J)
      ZR(J,K) = T1(J) + T2(J)
      S (J,2) = S(J,1)
   11 S (J,1) = SEGLP(J,KS) + ZR(J,K)
      CALL CROSS (WMEG(1,KS),V,T)
      IF  (KE.EQ.0)  GO TO 12
      CALL MAT31 (BD(7,KE),BAR(4,KI),T2)
      CALL DOT31 (D(1,1,KS),T2,T1)
   12 DO  13  J=1,3
      T(J) = T(J) + BAR(J+12,KI)
   13 E(J,3,K) = T1(J)
      CALL DOT31 (D(1,1,KS),T,V)
      DO  14  J=1,3
   14 V(J) = V(J) + SEGLV(J,KS)
      FB(K) = 0.0
      FP(K) = 0.0
      IF  (K.EQ.K1)  GO TO 20
      DO  15  J=1,3
   15 U(J,K-1) = S(J,1) - S(J,2)
      BL(K-1) = DSQRT(U(1,K-1)**2 + U(2,K-1)**2 + U(3,K-1)**2)
      DO  16  J=1,3
   16 U(J,K-1) = U(J,K-1)/BL(K-1)
      STRAIN = (BL(K-1)/BB(KNL-1)) - 1.0
      IF  (STRAIN.LT.0.0)  STRAIN = 0.0
      NT = NL(2,KNL)
      BLDOT = U(1,K-1)*(V(1)-R(1))
     *        + U(2,K-1)*(V(2)-R(2))
     *        + U(3,K-1)*(V(3)-R(3))
      STRDOT = (BB(KNL-1)*BLDOT-BL(K-1)*BBDOT(KNL-1))/BB(KNL-1)**2
      CALL FRCDFL (STRAIN,STRDOT,NT,0,FPK,ELOSS)
      CALL FRCDFL (STRAIN,STRDOT,NT,1,FBK,ELOSS)
      PTLOSS(1,K-1) = BB(KNL-1)*ELOSS
      FP(K-1) = FPK
      FB(K-1) = FBK
   20 CONTINUE
C
C     SECOND LOOP ON K
C     COMPUTE FCE(K),E1(K),E2(K),EDOT(K),FR(K),U1(KS),U2(KS)
C     NEED     FB(K&K-1),U(K&K-1),ZR(K),E3(K)
C
      DO  30  K=K1,K2
      KNL = KNLO + K
      KI = NL(1,KNL)
      KS    = IABS(IBAR(1,KI))
      IF  (KS.GT.100)  KS = MOD(KS,100)
      DO  21  J=1,3
      FCE(J,K) = FB(K)*U(J,K)
   21 IF  (K.NE.K1)  FCE(J,K) = FCE(J,K) - FB(K-1)*U(J,K-1)
      NT = IBAR(3,KI)
```

```
      NF = NTAB(NT+5)
      IF  (NF.EQ.0 .AND. IND.EQ.0)  GO TO 30
      IF  (IBAR(4,KI).EQ.0)  GO TO 22
      CALL DOT31 (D(1,1,KS),BAR(10,KI),T1)
      GO TO 24
   22 DO  23  J=1,3
      T1(J) = 0.0
      IF  (K.NE.K2)  T1(J) = U(J,K)
   23 IF  (K.NE.K1)  T1(J) = T1(J) + U(J,K-1)
   24 CALL CROSS (T1,E(1,3,K),E(1,1,K))
      CALL CROSS (E(1,3,K),E(1,1,K),E(1,2,K))
      DO  25  J=1,3
      EDOT(J,K) = DSQRT(E(1,J,K)**2 + E(2,J,K)**2 + E(3,J,K)**2)
      DO  25  I=1,3
   25 E(I,J,K) = E(I,J,K)/EDOT(J,K)
      CALL DOT31 (E(1,1,K),FCE(1,K),FR(1,K))
   30 CONTINUE
   31 CONTINUE
      IF  (NTP.LE.0)  GO TO 41
C
C     SUM FCE,FR FOR TIE-POINTS
C
      KNL1 = KNL0 + 2
      KNL2 = KNL0 + K2
      DO  40  KNL=KNL1,KNL2
      KI = NL(1,KNL)
      KS    = IABS(IBAR(1,KI))
      IF  (KS.LT.100)  GO TO 40
      KS1 = KS/100
      KH = KNL - KNL0
      MH = 0
      DO  38  JNL=KNL1,KNL
      KI = NL(1,JNL-1)
      KS    = IABS(IBAR(1,KI))
      IF  (KS.LT.100)  GO TO 38
      KS2 = KS/100
      IF  (KS2.NE.KS1)  GO TO 38
      JH = JNL-1 - KNL0
      IF  (MH.EQ.0)  MH = JH
      DO  37  J=1,3
      FCE(J,MH) = FCE(J,MH) + FCE(J,KH)
   37 FCE(J,JH) = FCE(J,MH)
      CALL DOT31 (E(1,1,JH),FCE(1,JH),FR(1,JH))
   38 CONTINUE
      IF  (MH.EQ.0)  GO TO 40
      KI = NL(1,KNL)
      IBAR(1,KI) = -IABS(IBAR(1,KI))
      DO  39  J=1,3
   39 FCE(J,KH) = FCE(J,MH)
      CALL DOT31 (E(1,1,MH),FCE(1,MH),FR(1,MH))
```

```
      CALL DOT31 (E(1,1,KH),FCE(1,KH),FR(1,KH))
   40 CONTINUE
C
C     IF CALL IS FROM SUBROUTINE CONTCT,
C     ADD FORCES (FCE) MODIFIED BY FRICTION TO U1,U2 ARRAYS.
C
   41 IF  (IND.NE.0)  GO TO 52
      K2 = 0
      DO  51  JB=J1,J2
      IF  (NPTPLY(JB).LE.0)  GO TO 51
      K1 = K2 + 1
      K2 = K2 + NPTPLY(JB)
      DO  50  K=K1,K2
      KNL = KNLO + K
      KI = NL(1,KNL)
      IF  (IBAR(1,KI).LT.0)  GO TO 50
      KS = IBAR(1,KI)
      IF  (KS.GT.100)  KS = MOD(KS,100)
      NT = IBAR(3,KI)
      NF = NTAB(NT+5)
      IF  (NF.EQ.0)  GO TO 43
      DO  42  J=1,3
   42 T1(J) = FR(J,K)
      FR1 = TAB(NF+2)*DABS(T1(3))
      FR2 = TAB(NF+4)*DABS(T1(3))
      IF  (DABS(T1(1)).GT.FR1)  T1(1) = DSIGN(FR1,T1(1))
      IF  (DABS(T1(2)).GT.FR2)  T1(2) = DSIGN(FR2,T1(2))
      CALL MAT31 (E(1,1,K),T1,FCE(1,K))
   43 CALL CROSS (ZR(1,K),FCE(1,K),T2)
      CALL MAT31 (D(1,1,KS),T2,T1)
      DO  44  J=1,3
      U1(J,KS) = U1(J,KS) + FCE(J,K)
   44 U2(J,KS) = U2(J,KS) + T1(J)
   50 CONTINUE
   51 CONTINUE
   52 KNLO = KNLO + K2
      CALL ELTIME (2,38)
      RETURN
      END
```

```
      SUBROUTINE HBPLAY
C                                                     REV 19 10/23/79
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36)
      COMMON/CNTSRF/ PL(17,30),BELT(20,8),TPTS(5,8),BD(24,40)
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)
      COMMON/HRNESS/ BAR(15,100),BB(100),BBDOT(100),PLOSS(2,100),
     *               XLONG(20),HTIME(2),IBAR(5,100),NL(2,100),
     *               NPTSPB(20),NPTPLY(20),NTHRNS(20),NBLTPH(5)
C     THIS COMMON/TEMPVS/ IS SHARED BY HPTURB, HBPLAY, HBELT AND HSETC.
      COMMON/TEMPVS/ B(3,3,3),S(3,3),T(3),R(3),V(3),T1(3),T2(3),
     *               E(3,3,50),EDOT(3,50),FCE(3,50),FR(3,50),ZR(3,50),
     *               TR(3,50),U(3,50),PTLOSS(2,50),DL(50),FB(50),FP(50),
     *               OLDBB(100),RHS(3,54),C(3,3,200),IJK(54,54)
      IF  (NHRNSS.LE.0)  GO TO 99
C
C     SAVE PREVIOUS NL,BB AND PLOSS ARRAYS.
C     USE IJK,OLDBB AND PTLOSS AS TEMP STORAGE.
C
      DO  10  I=1,100
      IJK(I,1) = NL(1,I)
      PTLOSS(I,1) = PLOSS(1,I)
   10 OLDBB(I) = BB(I)
      JNL = 1
      J1 = 1
      K1 = 1
      LL = 0
      DO  90  NH=1,NHRNSS
      IF  (NBLTPH(NH).LE.0)  GO TO 90
      J2 = J1 + NBLTPH(NH) - 1
      DO  80  NB=J1,J2
      L1 = LL
      IF  (NPTSPB(NB).LE.0)  GO TO 80
      K2 = K1 + NPTSPB(NB) - 1
      KB = 0
      DO  30  K=K1,K2
      KB = KB + 1
C
C     HERE K IS INDEX OF ALL POINTS
C          KB IS INDEX OF POINTS ON A SINGLE BELT
C          LL IS INDEX OF ALL POINTS IN PLAY
C          JB IS INDEX OF PREVIOUS POINT ON BELT IN PLAY
C
      KS = IABS(IBAR(1,K))
      IF  (KS.GT.100)  KS = MOD(KS,100)
      KE = IBAR(2,K)
      CALL DOT31 (D(1,1,KS),BAR(4,K),T1)
      CALL DOT31 (D(1,1,KS),BAR(7,K),T2)
```

```
      DO  11  J=1,3
   11 U(J,KB) = SEGLP(J,KS) + T1(J) + T2(J)
      IF  (K.EQ.K1)  GO TO 30
      LL = LL + 1
   12 JJ = NL(1,LL)
      JB = JJ - K1 + 1
      DSS = 0.0
      DO  13  J=1,3
      ZR(J,KB) = U(J,KB) - U(J,JB)
   13 DSS = DSS + ZR(J,KB)**2
      BL(LL) = DSQRT(DSS)
      IF  (JJ.EQ.K1 .OR. IABS(IBAR(1,JJ)).GT.100)  GO TO 30
      JS = IBAR(1,JJ)
      JE = IBAR(2,JJ)
      IF  (JE.LE.0)  GO TO 30
      CALL MAT31 (BD(7,JE),BAR(4,JJ),T2)
      CALL DOT31 (D(1,1,JS),T2,R)
      DPR = 0.0
      DO  17  J=1,3
   17 DPR = DPR + R(J)*(ZR(J,KB)/BL(LL) - ZR(J,JB)/BL(LL-1))
      IF  (DPR.LT.0.0)  GO TO 30
      LL = LL - 1
      GO TO 12
   30 NL(1,LL+1) = K
      L2 = L1 + 1
      LL = LL + 1
      L3 = LL-1
      DO  31  J=L2,LL
   31 NL(2,J) = NTHRNS(NB)
      IF  (XLONG(NB).EQ.0.0)  GO TO 35
C
C     FIRST TIME IN ROUTINE, SET INITIAL BB ARRAY.
C     INPUT XLONG MUST BE NON-ZERO TO TRIGGER THIS TEST.
C
      XLG = 0.0
      DO  32  J=L2,L3
   32 XLG = XLG + BL(J)
      XLG = 1.0 + XLONG(NB)/XLG
      DO  33  J=L2,L3
   33 BB(J) = XLG*BL(J)
      XLONG(NB) = 0.0
      GO TO 52
C
C     DETERMINE IF NEW NL ARRAY IS DIFFERENT FROM PREVIOUS NL ARRAY.
C     IF SO, RECOMPUTE BB ELEMENTS FOR POINTS THAT ARE DIFFERENT.
C
   35 IF  (NL(1,L2).EQ.IJK(JNL,1))  GO TO 61
      WRITE (6,62)
   62 FORMAT ('0 LOGIC ERROR IN SUB HBPLAY. PROGRAM TERMINATED.')
      STOP 42
```

```
 61   LTEST = 0
      M = L2
      N = JNL
 36   IF  (NL(1,M+1)-IJK(N+1,1))  39,37,41
 37   BB(M) = OLDBB(N)
      PLOSS(1,M) = PTLOSS(N,1)
 38   M = M+1
      N = N+1
      IF  (M-LL)  36,51,51
C
C     POINT M+1 IS NEW.
C
 39   MO = M
      NO = N
      LTEST = 1
 40   M = M+1
      GO TO 43
C
C      POINT N+1 IS DROPPED.
C
 41   MO = M
      NO = N
      LTEST = 1
 42   N = N+1
 43   IF  (NL(1,M+1)-IJK(N+1,1))  40,44,42
C
C     POINTS NO TO N+1 ARE BEING REPLACED WITH POINTS MO TO M+1.
C
 44   SUMBL = 0.0
      DO  45  J=MO,M
 45   SUMBL = SUMBL + BL(J)
      SUMPL = 0.0
      SUMBB = 0.0
      DO  46  J=NO,N
      SUMPL = SUMPL + PTLOSS(J,1)
 46   SUMBB = SUMBB + OLDBB(J)
      RATPL = SUMPL/SUMBL
      RATIO = SUMBB/SUMBL
      DO  47  J=MO,M
      PLOSS(1,J) = RATPL*BL(J)
 47   BB(J) = RATIO*BL(J)
      GO TO 38
 51   JNL = N+1
      IF  (LTEST.EQ.0)  GO TO 79
C
C     PRINT NEW POINT ARRAY IF DIFFERENT.
C
 52   NPTS = LL - L1
      WRITE  (6,53)  NH,NB,NPTS,NTHRNS(NB)
 53   FORMAT ('0 HBPLAY  NH,NB,NPTS,NT=',4I6)
```

```
      WRITE  (6,54)  (NL(1,J),J=L2,LL)
 54   FORMAT ('  NL(1)=',15I8/(8X,15I8))
      WRITE  (6,55)  (BB(J),J=L2,L3)
 55   FORMAT ('  BB   =',6X,14F8.3/(6X,15F8.3))
 79   K1 = K2 + 1
 80   NPTPLY(NB) = LL - L1
      J1 = J2 + 1
 90   CONTINUE
 99   RETURN
      END
```

```
      SUBROUTINE HINPUT
C                                                       REV 19 10/23/79
C     CONTROLS THE INPUT OF CARDS F.8.A - F.8.D CONTAINING THE SETUP AND
C     CONTROL OF THE HARNESS BELT SYSTEM.
C
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36)
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),
     *               UNITL,UNITM,UNITT,GRAVTY(3)
      COMMON/HRNESS/ BAR(15,100),BB(100),BBDOT(100),PLOSS(2,100),
     *               XLONG(20),HTIME(2),IBAR(5,100),NL(2,100),
     *               NPTSPB(20),NPTPLY(20),NTHRNS(20),NBLTPH(5)
      COMMON/TABLES/MXNTI,MXNTB,MXTB1,MXTB2,NTI(50),NTAB(500),TAB(2000)
      COMMON/CNTSRF/ PL(17,30),BELT(20,8),TPTS(6,8),BD(24,40)
      COMMON/TITLES/ DATE(3),COMENT(40),VPSTTL(20),BDYTTL(5),
     *               BLTTTL(5,8),PLTTL(5,30),BAGTTL(5,6),SEG(30),
     *               JOINT(30),CGS(30),JS(30)
      REAL DATE,COMENT,VPSTTL,BDYTTL,BLTTTL,PLTTL,BAGTTL,SEG,JOINT
      LOGICAL*1 CGS,JS
C     THIS COMMON/TEMTVS/ IS SHARED BY CINPUT, FINPUT, HINPUT AND FDINIT
      COMMON/TEMPVS/ JTITLE(5,51),NF(5),MS(3),KTITLE(31)
      REAL JTITLE,KTITLE
      IF  (NHRNSS.EQ.0)  GO TO 99
C
C     INPUT CARD F.8.A
C         (NOTE: NHRNSS NOW SUPPLIED ON INPUT CARD D.1)
C         NBLTPH - NO. OF BELTS PER HARNESS
C
      READ  (5,11)  (NBLTPH(I),I=1,NHRNSS)
   11 FORMAT(18I4)
      WRITE (6,12) NHRNSS,(NBLTPH(I),I=1,NHRNSS)
   12 FORMAT('1 HARNESS-BELT SYSTEM INPUT',93X,'CARDS F.8'//
     *       '  NO. OF HARNESSES =',I4//
     *       '  NO. OF BELTS PER HARNESS =',5I6)
      J1 = 1
      K1 = 1
      DO 90 I=1,NHRNSS
      IF (NBLTPH(I).LE.0) GO TO 90
      J2 = J1 + NBLTPH(I) -1
C
C     INPUT CARD F.8.B - NPTSPB - NO. OF POINTS PER BELT.
C
      READ (5,11) (NPTSPB(J),J=J1,J2)
      WRITE (6,13) I,(NPTSPB(J),J=J1,J2)
   13 FORMAT('0 FOR HARNESS NO.',I3,' NO. OF POINTS PER BELT =',20I4)
      DO 80 J=J1,J2
      IF (NPTSPB(J).EQ.0) GO TO 80
C
```

```
C     INPUT CARD F.8.C - 5 FUNCTION NOS AND LENGTH OF EACH BELT.
C
      READ  (5,14)      NF,XLONG(J)
   14 FORMAT(5I4,F12.6)
      WRITE (6,15) I,J,NF,XLONG(J),UNITL
   15 FORMAT('0 HARNESS NO.',I3,' BELT NO.',I3,' FUNCTION NOS.',5I6,
     *       '  REFERENCE SLACK = ',F9.3,1X,A4/)
      IF  (XLONG(J).EQ.0.0)  XLONG(J) = EPS(24)
      WRITE  (6,16)
   16 FORMAT ('0     K     KS     KE     NT     NPD    NDR     FUNCTION NOS.',
     *          66X,'CARDS F.8.D'/)
C
C     SET UP POINTERS IN NTAB AND INITIAL VALUES OF TAB FOR BELT J
C         AS WAS DONE FOR OTHER CONTACTS IN SUBROUTINE FINPUT.
C
      NTHRNS(J) = MXNTB+1
      CALL FDINIT
      K2 = K1 + NPTSPB(J) - 1
      DO 70 K=K1,K2
C
C     INPUT CARD F.8.D
C
      READ  (5,21)  KS,KE,NPD,NDR,NF, (BAR(L,K),L=1,3)
   21 FORMAT (9I4,3F12.0)
      READ  (5,22)  (BAR(L,K),L=7,12)
   22 FORMAT (6F12.0)
      IBAR(1,K) = KS
      IBAR(2,K) = KE
      IBAR(4,K) = NPD
      IBAR(5,K) = NDR
      IBAR(3,K) = MXNTB+1
      CALL FDINIT
      SQRER = 1.0
      IF  (KE.NE.0)  SQRER = DSQRT(XDY(BAR(1,K),BD(7,KE),BAR(1,K)))
      DO  26  L=1,3
      IF  (KE.NE.0)  BAR(L+6,K) = BD(L+3,KE)
   26 BAR(L+3,K) = BAR(L,K)/SQRER
      WRITE  (6,31)  K,(IBAR(L,K),L=1,5),NF
   31 FORMAT (11I6)
   70 CONTINUE
      WRITE  (6,71)  UNITL,UNITL,UNITL,UNITL
   71 FORMAT ('0',12X,'BASE REFERENCE (',      A4,')',
     *               7X,'ADJUSTED REFERENCE (', A4,')',
     *              11X,'OFFSET (',             A4,')',
     *              11X,'PREFERRED DIRECTION (',A4,')'/
     *          5X,'K', 4(8X,'X',8X,'Y',8X,'Z',3X) /)
      WRITE  (6,72)  (K,(BAR(L,K),L=1,12),K=K1,K2)
   72 FORMAT (I6,3X,3F9.3,3X,3F9.3,3X,3F9.3,3X,3F9.3)
      K1 = K2+1
   80 CONTINUE
```

```
      J1 = J2+1
   90 CONTINUE
      DO 92  K=1,100
      BBDOT(K) = 0.0
      DO 91  J=1,2
   91 PLOSS(J,K) = 0.0
      DO 92  J=1,3
   92 BAR(J+12,K) = 0.0
   99 RETURN
      END
```

```
      SUBROUTINE HPTURB
C                                                    REV 19 10/23/79
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36)
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),
     *               UNITL,UNITM,UNITT,GRAVTY(3)
      COMMON/CNTSRF/ PL(17,30),BELT(20,8),TPTS(6,8),BD(24,40)
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)
      COMMON/HRNESS/ BAR(15,100),BB(100),BBDOT(100),PLOSS(2,100),
     *               XLONG(20),HTIME(2),IBAR(5,100),NL(2,100),
     *               NPTSPB(20),NPTPLY(20),NTHRNS(20),NBLTPH(5)
C     THIS COMMON/TEMPVS/ IS SHARED BY HPTURB, HBPLAY, HBELT AND HSETC.
      COMMON/TEMPVS/ B(3,3,3),S(3,3),T(3),R(3),V(3),T1(3),T2(3),
     *               E(3,3,50),EDOT(3,50),FCE(3,50),FR(3,50),ZR(3,50),
     *               TR(3,50),U(3,50),PTLOSS(2,50),BL(50),FB(50),FP(50),
     *               OLDBB(100),RHS(3,54),C(3,3,200),IJK(54,54)
      DIMENSION BLOSS(2,20),HLOSS(2,5)
      EQUIVALENCE (BLOSS(1,1),C(1,1,1)) , (HLOSS(1,1),C(1,1,10))
      LOGICAL LAST
      DATA MAXITR/10/
      CALL ELTIME (1,39)
      CALL HBPLAY
      DHT = 0.0
      IF  (TIME.NE.0.0)  DHT = TIME - HTIME(1)
      HTIME(1) = TIME
      DO  11  J=1,100
      PTLOSS(J,1) = 0.0
      OLDBB(J) = BB(J)
      DO  11  I=1,3
   11 BAR(I,J) = BAR(I+3,J)
      TSEC = 1000.0*TIME
      IF  (NPRT(28).NE.0)  WRITE (6,12) TSEC,UNITL,UNITM,UNITL,
     *                                  UNITL,UNITM,UNITL,UNITM
   12 FORMAT('1   HARNESS BELT RESULTS FOR TIME =',F9.3,' MSEC.'///
     *        36X,'BELT STRAIN',68X,'PENETRATION'/
     *        '       POINT   POINT  SEGMENT LENGTH  ENERGY LOSS',5X,
     *        'REFERENCE POINT (',A4,')',13X,'BELT FORCES  (',A4,')',
     *        9X,'ENERGY LOSS'/
     *        '        NO.     INDEX     NO.    (',A4,')   (',2A4,')',7X,
     *        'X',8X,'Y',8X,'Z',13X,'X',10X,'Y',10X,'Z',8X,'(',2A4,')'/)
      J1 = 1
      K0 = 1
      KNL0 = 0
      DO  61  NH=1,NHRNSS
      IF  (NBLTPH(NH).LE.0)  GO TO 61
      ITER = 1
      KNL1 = KNL0
C
```

```
C      START OF    DO  59  ITER=1,MAXITR    LOOP
C
   13 NJ2 = 54
      DO  14  I=1,NJ2
      DO  14  J=1,NJ2
   14 IJK(I.J) = 0
      KNL0 = KNL1
      J2   = J1 + NBLTPH(NH) - 1
C
C     SET UP C AND IJK ELEMENTS FOR TIE-POINTS.
C
      NTP  = 0
      IJ   = 0
      KNLK = KNL0 + 1
      K1   = KNLK
      DO  18  NB=J1,J2
      IF  (NPTPLY(NB).LE.0)  GO TO 18
      K2   = K1 + NPTPLY(NB) - 1
      DO  17  KNL=K1,K2
      KI   = NL(1,KNL)
      KS   = IABS(IBAR(1,KI))
      IF  (KS.LT.100)  GO TO 17
      KS1  = KS/100
      DO  15  K=KNLK,KNL
      KK   = K
      KI   = NL(1,K)
      KS   = IABS(IBAR(1,KI))
      IF  (KS.LT.100)  GO TO 15
      KS2  = KS/100
      IF  (KS2.EQ.KS1)  GO TO 16
   15 CONTINUE
   16 IF  (KK.EQ.KNL)  GO TO 17
      NTP  = NTP + 1
      KK1 = KK  - KNL0
      KK2 = KNL - KNL0
      IJK(NTP,KK1) = IJ + 1
      IJK(NTP,KK2) = IJ + 3
      IJ   = IJ + 4
   17 CONTINUE
      K1   = K2 + 1
   18 CONTINUE
      IF  (NTP.EQ.0)  GO TO 23
      DO  20  I=1,NTP
      DO  19  J=1,K2
      JJ1 = K2+1-J
      IF  (IJK(I,JJ1).EQ.0)  GO TO 19
      JJ2 = JJ1 + NTP
      IJK(I,JJ2) = IJK(I,JJ1)
      IJK(JJ2,I) = IJK(I,JJ1) + 1
      IJK(I,JJ1) = 0
```

```
   19 CONTINUE
      DO  20  J=1,3
   20 RHS(J,I) = 0.0
      DO  22  K=1,IJ,2
      DO  22  J=1,3
      DO  21  I=1,3
      C(I,J,K  ) =  0.0
   21 C(I,J,K+1) =  0.0
      C(J,J,K  ) =  1.0
   22 C(J,J,K+1) = -1.0
   23 KNLO = KNL1
      CALL HBELT (J1,J2,KNLO,1)
      KHO  = 0
      KNLO = KNL1
      DO  24  NB=J1,J2
      IF  (NPTPLY(NB).LE.0)  GO TO 24
      NPTS = NPTPLY(NB)
      CALL HSETC (NPTS,KHO,KNLO,NTP,IJ)
      KHO = KHO + NPTS
      KNLO = KNLO + NPTS
   24 CONTINUE
      MJ2 = -(KHO+NTP)
      IF  (NPRT(28).LT.3)  GO TO 29
      NJ2 = -MJ2
      DO  25  J=1,NJ2
   25 WRITE  (6,26)  J,(RHS(I,J),I=1,3),(IJK(J,I),I=1,NJ2)
   26 FORMAT (I6,3F12.6,20I4/(42X,20I4))
      DO  27  KLM=1,IJ
   27 WRITE  (6,28)  KLM,((C(J,I,KLM),I=1,3),J=1,3)
   28 FORMAT (I6,9F12.6)
   29 CALL FSMSOL (C,RHS,IJK,MJ2,IJ,54,200)
      IF  (NPRT(28).LT.3)  GO TO 31
      DO  30  J=1,NJ2
   30 WRITE  (6,26)  J,(RHS(I,J),I=1,3),(IJK(J,I),I=1,NJ2)
   31 ONE = 1.0
      DELMAX = 0.0
      SCALE = 1.0
      DO  44  IT=1,2
      K1 = KO
      KH = 0
      KR = NTP
      DO  43  NB=J1,J2
      IF  (NPTPLY(NB).LE.0)  GO TO 43
      K2 = K1 + NPTPLY(NB) - 1
      DO  42  K=K1,K2
      KH = KH + 1
      KR = KR + 1
C
C     HERE K IS INDEX OF ALL POINTS IN PLAY
C          KH IS INDEX OF ALL POINTS IN PLAY ON A SINGLE HARNESS
```

```
C         KR IS INDEX OF RHS ARRAY ELEMENTS
C
      KI = NL(1,K)
      KS    = IABS(IBAR(1,KI))
      IF  (KS.GT.100)  KS = MOD(KS,100)
      IF  (IBAR(5,KI).EQ.0)  GO TO 32
      CALL MAT31 (D(1,1,KS),RHS(1,KR),R)
      GO TO 37
C
C     NOTE: ENDPOINTS (K = K1 & K2) MUST BE TYPE 5.
C
   32 CALL DOT31 (E(1,1,KH),RHS(1,KR),T1)
      IF  (IT.EQ.2)  GO TO 33
      DELMAX = DMAX1(DELMAX,DABS(T1(2)/DMIN1(BB(K),BB(K-1))))
      GO TO 34
   33 BB(K  ) = BB(K  ) + SCALE*T1(2)
      BB(K-1) = BB(K-1) - SCALE*T1(2)
   34 DO  35  J=1,3
   35 T2(J) = T1(1)*E(J,1,KH) + T1(3)*E(J,3,KH)
      CALL MAT31 (D(1,1,KS),T2,R)
      IF  (NPRT(28).GE.3)  WRITE (6,36) K,T1,T2,R
   36 FORMAT ('0',I6,3(3X,3F12.6))
   37 IF  (IT.EQ.2)  GO TO 39
      DO  38  J=1,3
   38 DELMAX = DMAX1(DELMAX,DABS(R(J)/DMAX1(EPS(1),DABS(BAR(J+3,KI)))))
      GO TO 42
   39 DO  40  J=1,3
   40 BAR(J+3,KI) = BAR(J+3,KI) + SCALE*R(J)
      KE = IBAR(2,KI)
      IF  (KE.EQ.0)  GO TO 42
      RER = XDY(BAR(4,KI),BD(7,KE),BAR(4,KI))
      IF  (RER.LE.1.0)  GO TO 42
      SQRER = 1.0/DSQRT(RER)
      DO  41  J=1,3
   41 BAR(J+3,KI) = SQRER*BAR(J+3,KI)
   42 CONTINUE
      K1 = K2 + 1
   43 CONTINUE
      IF  (IT.EQ.2)  GO TO 44
      IF  (DELMAX.NE.0.0)  SCALE = DMIN1(ONE,EPS(1)/DELMAX)
   44 CONTINUE
      IF  (NPRT(28).GE.2)  WRITE  (6,45)  ITER,DELMAX,SCALE
   45 FORMAT ('0  ITER =',I6,'  DELMAX =',F15.6,'  SCALE =',F15.6)
      LAST = DELMAX.LE.EPS(2) .OR. ITER.EQ.MAXITR
      IF  (.NOT.LAST)  GO TO 52
      KH = 0
      K1 = K0
      HLOSS(1,NH) = 0.0
      HLOSS(2,NH) = 0.0
      DO  51  NB=J1,J2
```

```
      BLOSS(1,NB) = 0.0
      BLOSS(2,NB) = 0.0
      IF  (NPTPLY(NB).LE.0)  GO TO 51
      K2 = K1 + NPTPLY(NB) - 1
      KK1 = NL(1,K1)
      KK2 = NL(1,K2)
      DO  46  K=KK1,KK2
      DO  46  J=1,3
   46 BAR(J+12,K) = 0.0
      IF  (DHT.EQ.0.0)  GO TO 49
      DO  48  K=K1,K2
      KH = KH + 1
      KI = NL(1,K)
      PLOSS(2,KI ) = PLOSS(2,KI ) + DHT*PTLOSS(2,KH)
      IF  (K.EQ.K1)  GO TO 47
      BBDOT(K-1) = (BB(K-1)-OLDBB(K-1))/DHT
      PLOSS(1,K-1) = PLOSS(1,K-1) + DHT*PTLOSS(1,KH-1)
      BLOSS(1,NB) = BLOSS(1,NB) + PLOSS(1,K-1)
   47 DO  48  J=1,3
   48 BAR(J+12,KI) = (BAR(J+3,KI)-BAR(J,KI))/DHT
      BBDOT(K2) = 0.0
      PLOSS(1,K2) = 0.0
   49 K1 = K2+1
      DO  50  K=KK1,KK2
   50 BLOSS(2,NB) = BLOSS(2,NB) + PLOSS(2,K)
      HLOSS(1,NH) = HLOSS(1,NH) + BLOSS(1,NB)
      HLOSS(2,NH) = HLOSS(2,NH) + BLOSS(2,NB)
   51 CONTINUE
   52 IF  (NPRT(28).EQ.0)  GO TO 59
      IF  (.NOT.LAST .AND. IABS(NPRT(28)).EQ.1)  GO TO 59
      K1 = K0
      KH = 0
      DO  57  NB=J1,J2
      IF  (NPTPLY(NB).LE.0)  GO TO 57
      WRITE (6,53)  NB,NH
   53 FORMAT ('0  BELT NO.',I4,' OF HARNESS NO.',I4)
      K2 = K1 + NPTPLY(NB) - 1
      DO  54  K=K1,K2
      KH = KH + 1
      KI = NL(1,K)
      KS = IBAR(1,KI)
      BK = 0.0
      IF  (K.NE.K1)  BK = BB(K-1)
      PLS = 0.0
      IF  (K.NE.K1)  PLS = PLOSS(1,K-1)
   54 WRITE (6,55) K,KI,KS,BK,PLS,(BAR(J,KI),J=4,6),
     *             (FCE(J,KH),J=1,3),PLOSS(2,KI)
   55 FORMAT (3I8,F10.3,F12.3,2X,3F9.3,3X,3F11.3,3X,F12.3)
      IF  (LAST)  WRITE (6,56)  BLOSS(1,NB),BLOSS(2,NB)
   56 FORMAT ('0    TOTAL BELT ENERGY LOSS',7X,F12.3,68X,F12.3)
```

```
      K1 = K2 + 1
   57 CONTINUE
      IF  (LAST)  WRITE (6,58)  HLOSS(1,NH),HLOSS(2,NH)
   58 FORMAT ('0 TOTAL HARNESS ENERGY LOSS',7X,F12.3,68X,F12.3)
   59 ITER = ITER + 1
C
C        END OF   DO  59  ITER=1,MAXITR   LOOP
C
      IF  (.NOT.LAST)  GO TO 13
      IF  (ITER.GT.MAXITR)  WRITE  (6,60)  MAXITR,TSEC,DELMAX,SCALE
   60 FORMAT ('0 HPTURB ITER =',I4,' AT TIME =',F8.3,
     *          ' MSEC.  DELMAX =',F10.6,'  SCALE =',F10.6)
      J1 = J2 + 1
      K0 = K1
   61 CONTINUE
      IF  (NPRT(28).LT.0)  NPRT(28) = 0
      CALL ELTIME (2,39)
      RETURN
      END
```

```
      SUBROUTINE HSETC (NPTS,KHO,KNLO,NTP,IJ)
C                                                       REV 19 10/30/79
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)
      COMMON/TABLES/ MXNTI,MXNTB,MXTB1,MXTB2,NTI(50),NTAB(500),TAB(2000)
      COMMON/HRNESS/ BAR(15,100),BB(100),BBDOT(100),PLOSS(2,100),
     *               XLONG(20),HTIME(2),IBAR(5,100),NL(2,100),
     *               NPTSPB(20),NPTPLY(20),NTHRNS(20),NBLTPH(5)
C     THIS COMMON/TEMPVS/ IS SHARED BY HPTURB, HBPLAY, HBELT AND HSETC.
      COMMON/TEMPVS/ B(3,3,3),S(3,3),T(3),R(3),V(3),T1(3),T2(3),
     *               E(3,3,50),EDOT(3,50),FCE(3,50),FR(3,50),ZR(3,50),
     *               TR(3,50),U(3,50),PTLOSS(2,50),BL(50),FB(50),FP(50),
     *               OLDBB(100),RHS(3,54),C(3,3,200),IJK(54,54)
      DIMENSION  KM(3),MK(2)
      ONE = 1.0
      KNL = KNLO
      KH = KHO
      K1 = KHO + NTP + 1
      K2 = KHO + NTP + NPTS
      DO  60  K=K1,K2
C
C     HERE K   IS INDEX OF IJK AND RHS ARRAYS
C          KH  IS INDEX OF POINTS IN PLAY ON EACH HARNESS
C          KNL IS INDEX OF ALL POINTS IN PLAY
C          KI  IS INDEX OF ALL POINTS
C
      KH = KH + 1
      KNL = KNL + 1
C
C     ZERO C(K,K) , C(K,K-1) , C(K,K+1) & RHS(K); SET IJK(K,K) = IJ
C
      KM(1) = K+1
      KM(2) = K-1
      KM(3) = K
      IF  (K.EQ.K2)  KM(1) = 0
      IF  (K.EQ.K1)  KM(2) = 0
      KK = IJ
      DO  12  L=1,3
      RHS(L,K) = 0.0
      IF  (KM(L).EQ.0)  GO TO 12
      KK = KK+1
      DO  11  I=1,3
      DO  11  J=1,3
   11 C(I,J,KK) = 0.0
   12 CONTINUE
      IJ = IJ+1
      IJK(K,K  ) = IJ
C
C     COMPUTE CNORM; IF ZERO, SET C(K,K) = I
```

```
C
      CNORM = 0.0
      IF  (K.NE.K2)  CNORM = FB(KH)/BL(KH)
      IF  (K.NE.K1)  CNORM = CNORM + FB(KH-1)/BL(KH-1)
      IF  (CNORM.NE.0.0)  GO TO 14
      KK = IJK(K,K)
      DO  13  I=1,3
   13 C(I,I,KK) = ONE
      GO TO 60
   14 KI = NL(1,KNL)
      KK = IBAR(3,KI)
      NFD = NTAB(KK+1)
      NFR = NTAB(KK+5)
C
C     SET UP B(3,3,3) AND S(3,3)
C
      MK(1) = KH
      MK(2) = KH-1
      IF  (K.EQ.K2)  MK(1) = 0
      IF  (K.EQ.K1)  MK(2) = 0
      DO  18 M=1,2
      KK = MK(M)
      IF  (KK.NE.0)  GO TO 16
      DO  15  I=1,3
      S(I,M) = 0.0
      DO  15  J=1,3
   15 B(I,J,M) = 0.0
      GO TO 18
   16 CALL DOT31 (E(1,1,KH),U(1,KK),T)
      KIM = KNL + I - M
      FB1 = FB(KK)/BL(KK)
      FB2 = FP(KK)/BB(KIM) - FB1
      FB3 = FP(KK)*BL(KK)/BB(KIM)**2
      DO  17  I=1,3
      SGN = ONE
      IF  (FR(I,KH).LT.0.0)  SGN = -ONE
      S(I,M) = SGN*(FB3*T(I))
      DO  17  J=1,3
   17 B(I,J,M) = SGN*(FB1*E(J,I,KH) + FB2*T(I)*U(J,KK))
   18 CONTINUE
      DO  19  I=1,3
      S(I,3) = -(S(I,1) + S(I,2))
      DO  19  J=1,3
   19 B(I,J,3) = -(B(I,J,1) + B(I,J,2))
      IF  (NFR.EQ.0)     GO TO 20
      R(1) = TAB(NFR+2)
      R(2) = TAB(NFR+4)
   20 R(3) = 0.0
      DO  50  M=1,3
      RH = 0.0
```

```
      IF  (M.EQ.3)       GO TO 31
      IF  (NFR.EQ.0)     GO TO 48
C
C     CONSTRAINTS 1 AND 2
C
      SGN    = -ONE
      FR3    = DABS(FR(M,KH)) - R(M)*DABS(FR(3,KH))
      IF  (IBAR(1,KI).GT.0)  RH = FR3
      IF  (FR3.LE.0.0)  GO TO 48
      GO TO 40
C
C     CONSTRAINT NO. 3
C
   31 IF  (NFD.EQ.0)     GO TO 48
      IF  (IBAR(1,KI).LT.0)  GO TO 40
      SGN    = ONE
      RMAG2 = TR(1,KH)**2 + TR(2,KH)**2 + TR(3,KH)**2
      RMAG   = DSQRT(RMAG2)
      RER2   = TR(1,KH)*E(1,3,KH)+ TR(2,KH)*E(2,3,KH)+ TR(3,KH)*E(3,3,KH)
      RER2   = EDOT(3,KH)*RER2
      RER    = DSQRT(RER2)
      PEN    = RMAG/RER - RMAG
      RRDOT  = BAR(4,KI)*BAR(13,KI)
     *       + BAR(5,KI)*BAR(14,KI)
     *       + BAR(6,KI)*BAR(15,KI)
      KS = IABS(IBAR(1,KI))
      IF  (KS.GT.100)  KS = MOD(KS,100)
      CALL DOT31 (D(1,1,KS),BAR(13,KI),T)
      ERDOT = E(1,3,KH)*T(1) + E(2,3,KH)*T(2) + E(3,3,KH)*T(3)
      C1     = PEN/RMAG2
      C2     = RMAG*EDOT(3,KH)/(RER*RER2)
      PDOT   = C1*RRDOT - C2*ERDOT
      CALL FRCDFL (PEN,PDOT,NFD,0,FDP,ELOSS)
      CALL FRCDFL (PEN,PDOT,NFD,1,FD ,ELOSS)
      RH     = FD - DABS(FR(3,KH))
      PTLOSS(2,KH) = ELOSS
      C1     = FDP*C1
      C2     = FDP*C2
      DO  32  J=1,3
   32 B(3,J,3) = B(3,J,3) - C1*TR(J,KH) + C2*E(J,3,KH)
   40 DO  47  LL=1,3
      L = 4 - LL
      IF  (KM(L).EQ.0)  GO TO 47
      DO  42  J=1,3
   42 V(J) = R(M)*B(3,J,L) + SGN*B(M,J,L)
      KL = KM(L)
      KML = KNL + KL - K
      KIL = NL(1,KML)
      IF  (IBAR(5,KIL).NE.0)  GO TO 43
      KHL = KH + KL - K
```

```
      CALL DOT31 (E(1,1,KHL),V,T)                                       HSETC  1510
      T(2) = R(M)*S(3,L) + SGN*S(M,L)                                   HSETC  1520
      CALL MAT31 (E(1,1,KHL),T,V)                                       HSETC  1530
   43 IF  (LL.NE.1)  GO TO 44                                           HSETC  1540
      VE = V(1)*E(1,M,KH) + V(2)*E(2,M,KH) + V(3)*E(3,M,KH)             HSETC  1550
      EV = DSIGN(ONE,VE)/DSQRT(V(1)**2+V(2)**2+V(3)**2)                 HSETC  1560
      IF  (IABS(I3AR(1,KI)).GT.100)  EV = 1.0                           HSETC  1570
      RH = EV*RH                                                        HSETC  1580
   44 IF  (IJK(K,KL).NE.0)  GO TO 45                                    HSETC  1590
      IJ = IJ+1                                                         HSETC  1600
      IJK(K,KL) = IJ                                                    HSETC  1610
   45 KK = IJK(K,KL)                                                    HSETC  1620
      DO  46  J=1,3                                                     HSETC  1630
      VEV = EV*V(J)                                                     HSETC  1640
      DO  46  I=1,3                                                     HSETC  1650
   46 C(I,J,KK) = C(I,J,KK) + E(I,M,KH)*VEV                             HSETC  1660
   47 CONTINUE                                                          HSETC  1670
      DO  41  I=1,3                                                     HSETC  1680
   41 RHS(I,K) = RHS(I,K) + RH*E(I,M,KH)                                HSETC  1690
      GO TO 50                                                          HSETC  1700
   48 KK = IJK(K,K)                                                     HSETC  1710
      DO  49  I=1,3                                                     HSETC  1720
      DO  49  J=1,3                                                     HSETC  1730
   49 C(I,J,KK) = C(I,J,KK) + E(I,M,KH)*E(J,M,KH)                       HSETC  1740
   50 CONTINUE                                                          HSETC  1750
   60 CONTINUE                                                          HSETC  1760
      RETURN                                                            HSETC  1770
      END                                                               HSETC  1780
```

```
      SUBROUTINE INITAL
C                                                   REV 19 05/25/79
C     PERFORMS CARD INPUT AND COMPUTATIONS FOR INITIAL
C     POSITIONING OF THE CRASH VICTIM'S BODY SEGMENTS.
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36)
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)
      COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(3,60),HA(3,60),HB(3,60),
     *               RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90),
     *               JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30)
      COMMON/VPOSTN/ ZPLT(3),SPLT(3),AXV(3,6),VATAB(6,101,6),
     *               VTO(6),VDT(6),TIMEV(6),OMEGV(6),NVTAB(6),INDXV(6)
      COMMON/TITLES/ DATE(3),COMENT(40),VPSTTL(20),BDYTTL(5),
     *               BLTTTL(5,8),PLTTL(5,30),BAGTTL(5,6),SEG(30),
     *               JOINT(30),CGS(30),JS(30)
      REAL DATE,COMENT,VPSTTL,BDYTTL,BLTTTL,PLTTL,BAGTTL,SEG,JOINT
      LOGICAL*1 CGS,JS
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),
     *               UNITL,UNITM,UNITT,GRAVTY(3)
      COMMON/TEMPVS/ TMP(18),WMGDEG(3,30),T(3),S(3),A(3,2),Z(3.3)
C              NOTE : CHAIN ALSO USES TEMPVS.
      DIMENSION  YPR(3,30) , IYPR(4,30)
C
C     INPUT CARD G.1.A (PLOT COORDINATES OF VEHICLE REFERENCE ORIGIN)
C
      READ(5,22) ZPLT,I1,J1,I2,J2,I3
   22 FORMAT(3F10.0,5I4)
      S(1) = 10.0
      S(2) =  6.0
      S(3) =  1.0
C
C     IF J1#0, INPUT CARD G.1.B (PLOT SCALING INPUT)
C
      IF  (J1.NE.0)  READ (5,22) S
      SPLT(1) = 1.0/S(3)
      SPLT(2) = 1.0/S(3)
      SPLT(3) = -(S(1)/S(2))/S(3)
      WRITE  (6,23)  ZPLT,I1,J1,I2,J2,I3,S
   23 FORMAT('1 SUBROUTINE INITAL INPUT',85X,'CARD G.1'//
     *   '      ZPLT(X)    ZPLT(Y)    ZPLT(Z)     I1     J1     I2     J2     I3',
     *   '      SPLT(1)    SPLT(2)    SPLT(3)'/3F10.0,5I6,3F10.2)
C
C     INPUT CARDS G.2.A - G.2.N
C
C     INITIAL LINEAR POSITION (IN) AND (IF I3=1) VELOCITY (IN/SEC)
C     OF EACH BASE BODY SEGMENT. IF I3=0, VELOCITY WILL BE SET TO
C     INITIAL VELOCITY OF VEHICLE. INPUTS IN INERTIAL REFERENCE.
```

```
C
      DO 37 J=1,NSEG
      IF(J.GT.1.AND.IABS(JNT(J-1)).GT.0) GO TO 37
      READ(5,24) (SEGLP(I,J),I=1,3),(SEGLV(I,J),I=1,3)
   24 FORMAT (6F10.0 , 4I3)
      IF(I3.GT.0) GO TO 37
      DO 36 I=1,3
   36 SEGLV(I,J) = SEGLV(I,NVEH)
   37 CONTINUE
C
C     INPUT CARDS G.3.A - G.3.N
C
C     FOR EACH BODY SEGMENT SUPPLY YAW, PITCH AND ROLL (DEGREES)
C     AND (IF I3=1) THE ANGULAR VELOCITY IN LOCAL REFERENCE (DEG/SEC).
C     IF I3=0, THE ANGULAR VELOCITY (BLANK ON INPUT CARDS) WILL BE SET
C     EQUAL TO THE INITIAL ANGULAR VELOCITY OF THE VEHICLE.
C
      FIRST = 0.0
      DO 40 J=1,NSEG
      READ (5,24) (YPR(I,J),I=1,3),(WMGDEG(I,J),I=1,3),(IYPR(I,J),I=1,4)
      ID1 = IYPR(1,J)
      DO 38 I=1,3
      IF (ID1.EQ.0) IYPR(I,J) = I
   38 WMEG(I,J) = WMGDEG(I,J)*RADIAN
      IF  (ID1.GE.0)  GO TO 60
C
C     READ CARD G.3.J2 FOR SEGMENT NO. J WHEN IYPR(1,J) IS NEGATIVE.
C
      READ  (5,24)  A,II,IK,JJ,JK
      IJ = II
      LK = IK
      DO  54  K=1,2
      IF  (IJ.GT.0)  GO TO 52
      DO  51  I=1,3
   51 Z(I,LK) = A(I,K)
      GO TO 53
   52 DA1 = A(1,K)*RADIAN
      DA2 = A(2,K)*RADIAN
      SA1 = DSIN(DA1)
      SA2 = DSIN(DA2)
      CA1 = DCOS(DA1)
      CA2 = DCOS(DA2)
      IJ1 = IJ+1
      IJ2 = IJ+2
      IF  (IJ1.GT.3)  IJ1= IJ1-3
      IF  (IJ2.GT.3)  IJ2= IJ2-3
      SGN = 1.0
      IF  (SA1.LT.0.0 .AND. CA2.LT.0.0)  SGN = -1.0
      Z(IJ ,LK) = SGN*SA1*CA2
      Z(IJ1,LK) = SGN*SA1*SA2
```

```
      Z(IJ2,LK) = SGN*CA1*CA2
   53 IJ = JJ
   54 LK = JK
      ZDOTIJ = Z(1,IK)*Z(1,JK) + Z(2,IK)*Z(2,JK) + Z(3,IK)*Z(3,JK)
      ZDOTII = Z(1,IK)*Z(1,IK) + Z(2,IK)*Z(2,IK) + Z(3,IK)*Z(3,IK)
      RATIO = ZDOTIJ/ZDOTII
      DO  55  I=1,3
   55 Z(I,JK) = Z(I,JK) - RATIO*Z(I,IK)
      LK = 6-IK-JK
      IT = MOD(JK-IK+3,3)
      IF  (IT.EQ.1)  CALL CROSS(Z(1,IK),Z(1,JK),Z(1,LK))
      IF  (IT.EQ.2)  CALL CROSS(Z(1,JK),Z(1,IK),Z(1,LK))
      DO  57  K=1,3
      IYPR(K,J) = 4-K
      SUM = 0.0
      DO  56  I=1,3
   56 SUM = SUM + Z(I,K)**2
      SQUM = DSQRT(SUM)
      DO  57  I=1,3
   57 D(K,I,J) = Z(I,K)/SQUM
      CALL YPRDEG (D(1,1,J),YPR(1,J))
      IF  (FIRST.EQ.0.0)  WRITE (6,58)
   58 FORMAT('0 INITIAL ANGULAR ROTATIONS COMPUTED FROM CARDS G.3.J2'//
     *         '  SEGMENT',10X,'SEGMENT PRIMARY AXIS',
     *       12X,'SEGMENT SECONDARY AXIS',30X,'ANGULAR ROTATIONS (DEG)'/
     *       '  NO. SEG',9X,'A1',8X,'A2',8X,'A3',11X,'B1',8X,'B2',8X,
     *       'B3',7X,'II  IK  JJ  JK',9X,'YAW',6X,'PITCH',5X,'ROLL'/)
      FIRST = 1.0
      WRITE (6,59) J,SEG(J),A,II,IK,JJ,JK,(YPR(I,J),I=1,3)
   59 FORMAT (I4,1X,A4,3X,3F10.3,3X,3F10.3,3X,4I4,3X,3F10.3)
   60 M = IYPR(4,J)
      IF (M.EQ.0) M=NGRND
      IF  (M.GE.J .AND. M.LE.NSEG)  STOP 24
      IF  (M.LT.0 .AND. -M.NE.IABS(JNT(J-1)))  STOP 25
      CALL DRCIJK (D,YPR,IYPR,HT,J)
      IF (I3.GT.0) GO TO 40
      CALL DOT31(D(1,1,NVEH),WMEG(1,NVEH),T)
      CALL MAT31(D(1,1,J),T,WMEG(1,J))
      DO 39 I=1,3
   39 WMGDEG(I,J) = WMEG(I,J)/RADIAN
   40 CONTINUE
      CALL VEHPOS
      CALL CHAIN
C
C     OUTPUT INITIAL BODY SEGMENT POSITIONS.
C
      WRITE (6,42) UNITL,UNITL,UNITT
   42 FORMAT('0 INITIAL POSITIONS (INERTIAL REFERENCE)',70X,'CARDS G.2'/
     *      /'  SEGMENT',11X,'LINEAR POSITION (',A4,')',
     *       14X,'LINEAR VELOCITY ('A4,'/',A4,')'/
```

```
     *       '  NO. SEG',2(9X,'X',11X,'Y',11X,'Z',5X) )
      WRITE (6,43) (J,SEG(J),(SEGLP(I,J),I=1,3),(SEGLV(I,J),I=1,3)
     *             ,J=1,NSEG)
   43 FORMAT(I4,1X,A4,3X,3F12.5,3X,3F12.5)
      WRITE (6,44) UNITT
   44 FORMAT('0 INITIAL ANGULAR ROTATION AND VELOCITY',71X,'CARDS G.3'//
     *       '  SEGMENT',11X,'ANGULAR ROTATION (DEG)'
     *       14X,'ANGULAR VELOCITY (DEG/',A4,')'/
     *       '  NO. SEG',8X,'YAW',8X,'PITCH',7X,'ROLL',
     *       13X,'X',11X,'Y',11X,'Z',15X,'IYPR'  )
      WRITE (6,46) (J,SEG(J),(YPR(I,J),I=1,3),(WMGDEG(I,J),I=1,3),
     *             (IYPR(I,J),I=1,4),J=1,NSEG)
   46 FORMAT(I4,1X,A4,3X,3F12.5,3X,3F12.5,3X,4I4)
      IF (I3.EQ.0) WRITE (6,45)
   45 FORMAT('0 LINEAR AND ANGULAR VELOCITIES HAVE BEEN SET EQUAL TO THE
     * INITIAL VEHICLE VELOCITIES.')
      IF  (NHRNSS.NE.0)  CALL HBPLAY
      IF  (I1.EQ.15)  CALL EQUILB (YPR,IYPR)
      CALL ELTIME(2,2)
      RETURN
      END
```

```
      SUBROUTINE KINPUT
C                                                       REV 19 09/18/79
C     PERFORMS THE FOLLOWING CARD INPUT AFTER CARDS E.1-E.4 (SUBROUTINE
C     CINPUT) AND BEFORE CARDS F.1-F.5 (SUBROUTINE FINPUT).
C         CARD E.5 - NWINDF: NO. OF WIND FORCE FUNCTIONS ON CARDS E.6
C                  - NJNTF : NO. OF JOINT FORCE FUNCTIONS ON CARDS E.7
C         CARDS E.6 - DEFINITIONS OF WIND FORCE FUNCTIONS
C         CARDS E.7 - DEFINITIONS OF JOINT RESTORING FORCE FUNCTIONS
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36)
      COMMON/TABLES/MXNTI,MXNTB,MXTB1,MXTB2,NTI(50),NTAB(500),TAB(2000)
      COMMON/TEMPVS/ JTITLE(5,51),NF(5),MS(3),KTITLE(31),TH(50)
C     NOTE: TEMPVS IS SHARED HERE WITH SUBROUTINES CINPUT AND FINPUT.
      REAL BLANK,JTITLE,KTITLE
      DATA BLANK/'    '/
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),
     *               UNITL,UNITM,UNITT,GRAVTY(3)
   11 FORMAT(2I6)
      J1 = MXTB1+1
      IF (NWINDF.LE.0) GO TO 31
      DO 30 K=1,NWINDF
C
C     INPUT CARD E.6.A - FUNCTION NO. AND TITLE
C
      READ (5,12) I,(KTITLE(J),J=1,5)
   12 FORMAT(I4,4X,5A4)
      WRITE (6,13) I,(KTITLE(J),J=1,5),I,J1
   13 FORMAT('1 WIND FORCE FUNCTION NO.',I4,4X,5A4,10X,'NTI(',I2,') =',
     *       I5,43X,'CARDS E.6'//)
      IF (I.LE.0.OR.I.GT.50) WRITE (6,14)
   14 FORMAT('0 IMPROPER FUNCTION NO. PROGRAM TERMINATED.')
      IF (I.LE.0.OR.I.GT.50) STOP 11
      IF (NTI(I).NE.0) WRITE (6,15) I
   15 FORMAT('0 FUNCTION NO.',I4,' HAS ALREADY BEEN INPUTTED AND WILL BE
     * REPLACED BY THIS FUNCTION.')
      NTI(I) = J1
      DO 16 J=1,5
   16 JTITLE(J,I) = KTITLE(J)
      J2 = J1+4
C
C     INPUT CARD E.6.B - D0 THRU D4 (FOR NOW A BLANK CARD)
C
      READ  (5,17) (TAB(J),J=J1,J2)
      WRITE (6,18) (TAB(J),J=J1,J2)
   17 FORMAT(6F12.0)
   18 FORMAT(10X,'D0',13X,'D1',13X,'D2',13X,'D3',13X,'D4'/5F15.4//)
      J1 = J2+1
C
```

```
C     INPUT CARD E.6.C - NTMPTS
C
      READ  (5,11) NTMPTS
      WRITE (6,19) NTMPTS
   19 FORMAT('0 WIND FORCE TABLES FOR ',I6,' TIME POINTS.'//
     *        11X,'T',14X,'FX(T)',15X,'FY(T)',15X,'FZ(T)'  /)
      TAB(J1) = NTMPTS
      J1 = J1+1
      J2 = J1+4*NTMPTS-1
C
C     INPUT CARDS E.6.D-E.6.N - NTMPTS CARDS OF T,FX(T),FY(T),FZ(T)
C
      READ  (5,20) (TAB(J),J=J1,J2)
      WRITE (6,21) (TAB(J),J=J1,J2)
   20 FORMAT(4F12.0)
   21 FORMAT(3X,F12.6,3G20.6)
      J1 = J2+1
   30 CONTINUE
   31 IF (NJNTF.LE.0) GO TO 51
      DO 50 K=1,NJNTF
C
C     INPUT CARD E.7.A - FUNCTION NO. AND TITLE
C
      READ  (5,12) I,(KTITLE(J),J=1,5)
      WRITE (6,32) I,(KTITLE(J),J=1,5),I,J1
   32 FORMAT('1 JOINT FORCE FUNCTION NO.',I4,4X,5A4,10X,'NTI(',I2,') =',
     *          I5,42X,'CARDS E.7'//)
      IF (I.LE.0.OR.I.GT.50) WRITE (6,14)
      IF (I.LE.0.OR.I.GT.50) STOP 12
      IF (NTI(I).NE.0) WRITE (6,15) I
      NTI(I) = J1
      DO 33 J=1,5
   33 JTITLE(J,I) = KTITLE(J)
C
C     INPUT CARD E.7.B - D0,D1,D2,D3,D4 (FOR NOW A BLANK CARD).
C
      J2 = J1+4
      READ  (5,17) (TAB(J),J=J1,J2)
      WRITE (6,18) (TAB(J),J=J1,J2)
      J1 = J2+1
C
C     INPUT CARD E.7.C - NTHETA,NPHI
C
      READ  (5,11) NTHETA,NPHI
      TAB(J1  ) = NTHETA
      TAB(J1+1) = NPHI
      J1 = J1+2
      IF (NTHETA.LT.0) GO TO 38
      DO 35 J=1,NTHETA
   35 TH(J) = DFLOAT(J-1)*180.0/DFLOAT(NTHETA-1)
```

```
      WRITE (6,36) NTHETA,NPHI,(TH(J),J=2,NTHETA)
   36 FORMAT('0 FUNCTION IS TABULAR FOR' ,I3,' X',I3,' VALUES OF THETA A
     *ND PHI'//30X,'THETA'/5X,'PHI',5X,'THETAO',F16.3,4F20.3/
     *      (15X,5F20.3))
   37 FORMAT(F9.2,F10.3,5G20.7/(19X,5G20.7))
      GO TO 40
   38 NPOLY = -NTHETA -1
      WRITE (6,39) NPOLY,NPHI,(BLANK,J,J=1,NPOLY)
   39 FORMAT('0 FUNCTION IS COEFFICIENTS OF' ,I3,' ORDER POLYNOMIALS IN
     *(THETA-THETAO) FOR',I3,' VALUES OF PHI.'//
     *  27X,'COEFFICIENTS OF (THETA-THETAO)**N'/
     *  5X,'PHI',5X,'THETAO',7X,5(A4,'N =',I2,11X)/(26X,A4,'N =',I2,11X,
     *  A4,'N =',I2,11X,A4,'N =',I2,11X,A4,'N =',I2,11X,A4,'N =',I2)  )
   40 WRITE (6,21)
      DO 49 I=1,NPHI
      PHIDEG = DFLOAT(I-1)*360.0/DFLOAT(NPHI) - 180.0
C
C     INPUT CARDS E.7.D - E.7.N NPHI SETS WITH NTHETA ITEMS PER SET.
C        EACH SET I IS FOR PHI(I) = -180 +(I-1)*360/NPHI DEGREES AND
C        ASSUMES DATA FOR PHI(NPHI+1) = 180 IS SAME AS PHI(1) = -180.
C
      J2 = J1 + IABS(NTHETA) -1
      READ  (5,17) (TAB(J),J=J1,J2)
      WRITE (6,37) PHIDEG,(TAB(J),J=J1,J2)
      IF (NTHETA.LT.0) TAB(J1) = TAB(J1)*RADIAN
      IF (NTHETA.LT.0) GO TO 49
C
C     FOR TABULAR DATA, FILL IN ZERO VALUES WITH INTERPOLATED NEGATIVE
C     VALUES. OVERWRITE VALUE IN FIRST COLUMN (SUPPLIED AS THETAO) WITH
C     VALUE FOR THETA = 0 AND ALL OTHER ZERO VALUES.
C
      THETAO = TAB(J1)
      IF (THETAO.EQ.0.0) GO TO 49
      JJ = THETAO*DFLOAT(NTHETA-1)/180.0 + 1.0 + EPS(6)
      JJ1 = J1+JJ
      IERROR = 0
      IF (JJ1.GT.J2) IERROR = 1
      IF (TAB(JJ1).LE.0.0) IERROR = 2
      IF (IERROR.NE.0) GO TO 46
      DO 45 J=1,JJ
      J1J = J1+J-1
      IF (J.NE.1.AND.TAB(J1J).GT.0.0) IERROR = 3
   45 TAB(J1J) = TAB(JJ1)*(TH(J)-THETAO)/(TH(JJ+1)-THETAO)
   46 IF (IERROR.NE.0) WRITE (6,47) IERROR
   47 FORMAT('0 INPUT ERROR. INCONSISTENT VALUE OF THETAO. IERROR =',I2,
     *        ' PROGRAM TERMINATED.')
      IF (IERROR.NE.0) STOP 13
   49 J1 = J2+1
   50 CONTINUE
   51 MXTB1 = J1-1
```

```
      RETURN                                                                KINPUT  1510
      END                                                                   KINPUT  1520
```

```
      SUBROUTINE SPLINE (X,Y,F,N,L)
C                                                    REV 19 05/14/79
C
C     ROUTINE TO FIT A SET OF POLYNOMIALS OF DEGREE L
C     TO A SET OF GIVEN DATA POINTS (X(I),Y(I),I=1,N)
C
C     FUNCTION IS OF FORM:
C
C       Y = F(2,K) + F(3,K)*DX + F(4,K)*DX**2 + F(5,K)*DX**3
C
C       WHERE: DX = XX - F(1,K)
C              F(1,K) .LE. XX .LT. F(1,K+1)  ;  (SETS K)
C              IF  (XX.GT.F(1,N))  ;  USE K=N, CONSTANT FIT TO Y(N)
C              IF  (XX.LT.F(1,1))  ;  EXTRAPOLATED FIT FOR K=1
C
C                 F(1,I) = X(I) ,                I=1,N
C                 F(2,I) = Y(I) ,                I=1,N
C
C      DEGREE L                                          CONTINUITY
C             0   F(3,I) = F(4,I) = F(5,I) = 0 , I=1,N       NONE
C             1   F(4,I) = F(5,I) = 0  ,         I=1,N       Y
C             2   F(5,I) = 0 ,                   I=1,N       Y,Y'
C             3   CUBIC SPLINE                               Y,Y',Y''
C
C                 F(K,N)=0 FOR K=3,5 IN ALL CASES
C
C     FOR L=2 AND L=3 THE CHANGES IN THE L'TH DERIVATIVES ARE MINIMIZED
C
C     SPECIAL CASES:
C         N=1 ;     TREATED AS L=0
C         N=2 ;     TREATED AS L=MIN(L,1)
C         L<0 ;     TREATED AS L=0
C         L>3 ;     TREATED AS L=3
C
C     STORAGE REQUIRED  X(N),Y(N),F(5,N); SET BY CALLING PROGRAM
C
C     USAGE:
C       ALL COMPUTATIONS AND REAL VARIABLES ARE DOUBLE PRECISION
C       GIVEN:  L,N, (X(I),Y(I),I=1,N)
C       CALL SPLINE (X,Y,F,N,L)          ; SETS F
C
CC
CC      TO EVALUATE FUNCTION AND DERIVATIVES AT POINT XX
CC
C           DO  10  K=1,N
C           IF (K.EQ.N)  GO TO 11
C           IF (XX.LT.F(1,K+1)) GO TO 11
C       10  CONTINUE
C       11  DX = XX - F(1,K)
C           YY = F(2,K) + DX*(F(3,K)+DX*(F(4,K)+DX*F(5,K)))
```

```
C           YD = F(3,K) + DX*(2.0*F(4,K)+3.0*DX*F(5,K))
C           YDD = 2.0*F(4,K) + 6.0*DX*F(5,K)
C           YDDD = 6.0*F(5,K)
C           YDDDD = 0.0
CC
CC        FUNCTIONAL VALUE IN YY, DERIVATIVES IN YD'S
CC        REPEAT FOR NEXT VALUE OF XX
C
C       AUTHOR: DR. JOHN T. FLECK
C
        IMPLICIT REAL*8 (A-H,O-Z)
        DIMENSION X(N),Y(N),F(5,N),C(2,3)
        DO  20  I=1,N
        F(1,I) = X(I)
        DO  10  K=2,5
   10   F(K,I) = 0.0
        IF  (L.LT.3)  F(2,I) = Y(I)
   20   IF  (L.GT.0 .AND. I.LT.N)  F(3,I) = (Y(I+1)-Y(I))/(X(I+1)-X(I))
        IF  (L.LT.2 .OR.  N.LT.3)  GO TO 99
        IF  (L.GE.3)  GO TO 50
        D1 = X(2) - X(1)
        SS = 0.0
        DS = 0.0
        DO  30  I=3,N
        F(4,I-1) = F(3,I-1) - F(3,I-2) - F(4,I-2)
        DX1 = X(I) - X(I-1)
        DX2 = X(I-1) - X(I-2)
        DD = D1/DX1 + D1/DX2
        SS = SS + DD*DD
        DS = DS + DD*(F(4,I-1)/DX1 - F(4,I-2)/DX2)
   30   D1 = -D1
        F(4,1) = DS/SS
        DX = (X(2)-X(1))*F(4,1)
        F(3,1) = F(3,1) - DX
        DO  40  I=3,N
        XX = F(4,I-1) - DX
        F(3,I-1) = F(3,I-1) - XX
        F(4,I-1) = XX/(X(I)-X(I-1))
   40   DX = -DX
        GO TO 99
C
C       CUBIC SPLINE
C
   50   DO  51  I=2,N
        IF  (I.EQ.N)  GO TO 51
        F(4,I) = 3.0*(F(3,I)-F(3,I-1))
        F(5,I) = 2.0*(X(I+1)-X(I-1))
   51   F(3,I-1) = 0.0
        F(2,N) = -1.0
        F(3,1) = -1.0
```

```
      DO  60  I=3,N
      DX = X(I-1) - X(I-2)
      IF  (I.GT.3)  DX = DX/F(5,I-2)
      DO  60  K=3,5
50    F(K,I-1) = F(K,I-1)- F(K,I-2)*DX**((K-1)/2)
      DO  70  I=3,N
      NI = N-I
      DX = X(NI+3) - X(NI+2)
      DO  70  K=2,4
70    F(K,NI+2) = (F(K,NI+2) - DX*F(K,NI+3))/F(5,NI+2)
      DO  71  J=1,2
      DO  71  K=J,3
      C(J,K) = 0.0
      DO  71  I=3,N
      DX1 = X(I) - X(I-1)
      DX2 = X(I-1) - X(I-2)
71    C(J,K) = C(J,K) + (  (F(J+1,I  )-F(J+1,I-1))/DX1
     *                    - (F(J+1,I-1)-F(J+1,I-2))/DX2)
     *                  * (  (F(K+1,I  )-F(K+1,I-1))/DX1
     *                    - (F(K+1,I-1)-F(K+1,I-2))/DX2)
      DEN    =  C(1,1)*C(2,2) - C(1,2)*C(1,2)
      F(4,1) = (C(1,1)*C(2,3) - C(1,2)*C(1,3))/DEN
      F(4,N) = (C(2,2)*C(1,3) - C(1,2)*C(2,3))/DEN
      DO  72  I=3,N
72    F(4,I-1) = F(4,I-1) - F(4,1)*F(3,I-1) - F(4,N)*F(2,I-1)
      D1 = X(2) - X(1)
      F(3,1) = (Y(2)-Y(1))/D1 - (2.0*F(4,1)+F(4,2))*D1/3.0
      F(2,1) = Y(1)
      DO  80  I=2,N
      F(2,I) = Y(I)
      DX = X(I) - X(I-1)
      IF  (I.LT.N)  F(3,I) = F(3,I-1) + (F(4,I)+F(4,I-1))*DX
80    F(5,I-1) = (F(4,I)-F(4,I-1))/(3.0*DX)
      F(4,N) = 0.0
99    RETURN
      END
```

```
      SUBROUTINE UPDATE(I)
C                                                       REV 19 10/23/79
C     CALLED BY SUBROUTINE DINT
C
C       (I=1)  AT THE START OF A NEW STEP TO SETUP ANY NEW CONDITIONS
C              TO BE VALID FOR ENTIRE INTEGRATION STEP
C                A. UPDATE FORCE DEFLECTION FUNCTIONS(SUBROUTINE UPDFDC)
C                B. TEST FOR LOCKED JOINTS
C       NOTE: ARGUMENT I WILL BE SET TO -1 TO RESET INTEGRATOR.
C
C       (I=2)  AT THE END OF EACH SUCCESSFUL INTEGRATION STEP TO
C              COMPLETE CALCULATIONS FOR OUTPUT (SUBROUTINE AIRBG3).
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36)
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)
      COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(3,60),HA(3,60),HB(3,60),
     *               RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90),
     *               JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30)
      COMMON/CMATRX/ V1(3,30),V2(3,30),V3(3,12),B12(3,3,60),A22(3,3,60),
     *               F(3,30),TQ(3,30),WJ(30)
      COMMON/JBARTZ/ MNPL(    30),MNBLT(    8),MNSEG(    30),MNBAG(     6),
     *               MPL(3,5,30),MBLT(3,5,8),MSEG(3,5,30),MBAG(3,10,6),
     *               NTPL( 5,30),NTBLT( 5,8),NTSEG( 5,30)
      COMMON/TABLES/MXNTI,MXNTB,MXTB1,MXTB2,NTI(50),NTAB(500),TAB(2000)
      COMMON/FORCES/ PSF(7,30),BSF(4,20),SSF(10,20),BAGSF(3,20),
     *               PRJNT(6,30),NPANEL(5),NPSF,NBSF,NSSF,NBGSF
      COMMON/CSTRNT/ A13(3,3,24),A23(3,3,24),B31(3,3,24),B32(3,3,24),
     *               HHT(3,3,12),RK1(3,12),RK2(3,12),QQ(3,12),TQQ(3,12),
     *               RQQ(3,12),HQQ(3,12),SQQ(12),CFQQ(12),
     *               KQ1(12),KQ2(12),KQTYPE(12)
      COMMON/TEMPVI/ CREST,TTI(3),R1I(3),R2I(3),JSTOP(4,2,30)
      COMMON/CEULER/ IEULER(30),HIR(3,3,30),ANG(3,30),ANGD(3,30),
     *               FE(3,30),TQE(3,30),CONST(3,30)
      COMMON/HRNESS/ BAR(15,100),BB(100),BBDOT(100),PLOSS(2,100),
     *               XLONG(20),HTIME(2),IBAR(5,100),NL(2,100),
     *               NPTSPB(20),NPTPLY(20),NTHRNS(20),NBLTPH(5)
      DIMENSION TOTEST(3),LOCK(8,3) ,T(3)
      DATA LOCK/-8, 6, 5, 7,-3,-2,-4, 1,
     *           6,-8, 4,-3, 7,-1,-5, 2,
     *           5, 4,-8,-2,-1, 7,-6, 3/
C
C     CALL AIRBG3 FOR AIRBAG, IF ANY.
C
      IF  (NBAG.NE.0)  CALL AIRBG3(I)
      IF  (I.EQ.2)  GO TO 42
      CALL ELTIME (1,7)
      IF  (NPL.LE.0)  GO TO 13
```

```
C
C     CALL UPDFDC FOR EACH ALLOWED PLANE-SEGMENT CONTACT.
C
      NPSF = 0
      DO 12 J=1,NPL
      NK = MNPL(J)
      IF (NK.LE.0) GO TO 12
      DO 11 K = 1, NK
      NPSF = NPSF+1
      NT = NTPL(K,J)
      NF = NTAB(NT+5)
      CALL UPDFDC(NT)
      IF (NT.GT.0.OR.TAB(NF+3).EQ.0.0) GO TO 11
      CALL IMPULS(1,K,J)
      I = -1
   11 CONTINUE
   12 CONTINUE
   13 IF (NBLT.LE.0) GO TO 16
C
C     CALL UPDFDC FOR EACH ALLOWED BELT-SEGMENT CONTACT.
C
      DO 15 J=1,NBLT
      NK = MNBLT(J)
      IF (NK.LE.0) GO TO 15
      DO 14 K = 1,NK
      NT = NTBLT(K,J)
      NF = NTAB(NT+5)
      NT6 = NT+6
      CALL UPDFDC(NT)
C
C        AND FOR 2ND FUNCTION, IF FULL BELT FRICTION.
C
   14 IF (NF.NE.0) CALL UPDFDC(NT6)
   15 CONTINUE
C
C     CALL UPDFDC FOR EACH ALLOWED SEGMENT-SEGMENT CONTACT.
C
   16 NSSF = 0
      DO 18 J=1,NSEG
      NK = MNSEG(J)
      IF (NK.LE.0) GO TO 18
      DO 17 K = 1,NK
      NSSF = NSSF+1
      NT = NTSEG(K,J)
      NF = NTAB(NT+5)
      CALL UPDFDC(NT)
      IF (NT.GT.0.OR.TAB(NF+3).EQ.0.0) GO TO 17
      CALL IMPULS(3,K,J)
      I = -1
   17 CONTINUE
```

```
   18 CONTINUE
      IF (NHRNSS.LE.0) GO TO 71
C
C     CALL UPDFDC FOR EACH BELT OF HARNESS-BELT SYSTEMS.
C
      CALL HPTURB
      J1 = 1
      K1 = 1
      DO 70 II=1,NHRNSS
      IF (NBLTPH(II).LE.0) GO TO 70
      J2 = J1 + NBLTPH(II) - 1
      DO 69 J=J1,J2
      IF  (NPTPLY(J).LE.0)  GO TO 69
      NT = NTHRNS(J)
      CALL UPDFDC(NT)
      K2 = K1 + NPTPLY(J) - 1
      DO  68  K=K1,K2
      KI = NL(1,K)
      NT = IBAR(3,KI)
      CALL UPDFDC(NT)
   68 CONTINUE
      K1 = K2+1
   69 CONTINUE
      J1 = J2+1
   70 CONTINUE
   71 IF (NJNT.LE.0) GO TO 37
C
C            CHECK FOR IMPULSE ON JOINT STOPS
C     TO BE CALLED IF IN JOINT STOP (JSTOP(1)=1) THIS TIME STEP
C     BUT NOT IN IN JOINT STOP (JSTOP(2)=0) AT PREVIOUS TIME.
C
      DO 21 K=1,NJNT
      IF (JNT(K).EQ.0) GO TO 21
      IF (IABS(IPIN(K)).NE.4 .AND. VISC(7,3*K-2).EQ.0.0) GO TO 20
      DO 19 J=1,3
      K3J = 3*K-3+J
      IF (IABS(IPIN(K)).NE.4) K3J=3*K-2
      IF (IABS(IPIN(K)).EQ.4 .AND. VISC(7,K3J).EQ.0.0) GO TO 19
      IF (JSTOP(J,1,K).NE.1.OR.JSTOP(J,2,K).NE.0) GO TO 19
      CALL IMPULS(4,J,K)
      I = -1
   19 JSTOP(J,2,K) = JSTOP(J,1,K)
   20 IF (IGLOB(K).EQ.0) GO TO 21
      NT = IGLOB(K)
      MT = NTAB(NT+5)
      NT1 = NTAB(NT+2)
      NTAB(NT+2) = 0
      CALL UPDFDC(NT)
      NT = IABS(NT)
      NTAB(NT+2) = NT1
```

```
      IF (TAB(MT+3).EQ.0.0) GO TO 21
      IF (JSTOP(4,1,K).NE.1.OR.JSTOP(4,2,K).NE.0) GO TO 21
      CALL IMPULS(4,4,K)
      I = -1
   21 JSTOP(4,2,K) = JSTOP(4,1,K)
C
C     TEST TO LOCK OR UNLOCK JOINTS
C
C
C     CONDITIONS TO CHANGE SIGN OF IPIN(J)
C
C                          PINNED                UNPINNED
C         LOCKED (-1) IH.TQI > T1      (-2) ITQI > T1
C
C       UNLOCKED  (+1) IH.TQI < T2      (+2) ITQI < T2
C                              OR                   OR
C                          WJ  < T3             WJ  < T3
C
      DO 28 J=1,NJNT
      IF (IABS(IPIN(J)).EQ.4) GO TO 28
      IF (IPIN(J)) 22,28,23
   22 T1 = VISC(4,3*J-2)
      IF (T1.EQ.0.0) GO TO 28
      IF  (IPIN(J).NE.-1)  GO TO 51
      TQM = XDY(HB(1,2*J),D(1,1,J+1),TQ(1,J))
      ABSTQM = DABS(TQM)
      IF  (ABSTQM.GT.T1)  HA(2,2*J-1) = TQM
      TQM = ABSTQM
      GO TO 52
   51 TQM = DSQRT(TQ(1,J)**2 + TQ(2,J)**2 + TQ(3,J)**2)
      IF  (TQM.GT.T1)  CALL DOT31(HIR(1,1,J),TQ(1,J),HA(1,2*J-1))
   52 IF  (TQM-T1)  28,28,26
   23 T2 = VISC(5,3*J-2)
      IF  (HA(2,2*J).NE.0.0)  GO TO 54
      DO  53  K=1,3
   53 HA(K,2*J-1) = 0.0
   54 IF  (T2.EQ.0.0)  GO TO 24
      IF (IPIN(J).GE.2)  TQM = DSQRT(TQ(1,J)**2+TQ(2,J)**2+TQ(3,J)**2)
      IF (IPIN(J).EQ. 1) TQM = DABS(XDY(HB(1,2*J),D(1,1,J+1),TQ(1,J)))
      IF (TQM-T2) 25,28,28
   24 T3 = VISC(6,3*J-2)
      IF (T3.EQ.0.0) GO TO 28
      IF (WJ(J)-T3) 25,28,28
   25 CALL IMPLS2(0,J,HB(1,2*J))
      I = -1
   26 IPIN(J) = -IPIN(J)
      TMSEC = 1000.0*TIME
      IPINJ = -IPIN(J)
      WRITE (6,27) TMSEC,J,IPINJ,IPIN(J)
   27 FORMAT('0 AT TIME =',F9.3,' MSEC,   IPIN(',I2,') HAS BEEN CHANGED
```

```
     *FROM',I3,' TO',I3)
   28 CONTINUE
C
C      TEST TO LOCK OR UNLOCK EULER JOINTS AXES.
C      USE SAME TEST AS ABOVE BUT ON EACH AXIS SERARATELY.
C
C       IF LOCK(IEULER,K) IS NEGATIVE, AXIS K IS LOCKED;
C          TO UNLOCK AXIS SET IEULER TO -LOCK(IEULER,K).
C
C      IF LOCK(IEULER,K) IS POSITIVE, AXIS K IS UNLOCKED;
C        TO LOCK AXIS SET IEULER TO LOCK(IEULER,K).
C
      DO 36 J=1,NJNT
      IF (IABS(IPIN(J)).NE.4) GO TO 36
      JEULER = IEULER(J)
      CALL DOT31(HIR(1,1,J),TQ(1,J),TQTEST)
      DO 31 K=1,3
      K3J = 3*J-3+K
      NLOCK = LOCK(JEULER,K)
      IF (NLOCK.GT.0) GO TO 29
      IF (VISC(4,K3J).EQ.0.0) GO TO 31
      IF (DABS(TQTEST(K)).LE.VISC(4,K3J)) GO TO 31
      JEULER = -NLOCK
      HA(K,2*J-1) = TQTEST(K)
      GO TO 31
   29 IF  (HA(K,2*J).EQ.0.0)  HA(K,2*J-1) = 0.0
      IF (VISC(5,K3J).EQ.0.0) GO TO 30
      IF (DABS(TQTEST(K)).LT.VISC(5,K3J)) JEULER =  NLOCK
      GO TO 31
   30 IF (VISC(6,K3J).EQ.0.0) GO TO 31
      IF (DABS(ANGD(K,J)).LT.VISC(6,K3J)) JEULER =  NLOCK
   31 CONTINUE
      IF (JEULER.EQ.IEULER(J)) GO TO 36
      TMSEC = 1000.0*TIME
      WRITE (6,32) TMSEC,J,IEULER(J),JEULER
   32 FORMAT('0 AT TIME =',F9.3,' MSEC, IEULER(',I2,') HAS BEEN CHANGED
     *FROM',I3,' TO',I3)
      IF (JEULER.EQ.8)         GO TO 35
      IF (IEULER(J).EQ.7) GO TO 35
      IF (IEULER(J).EQ.6 .AND. (JEULER.EQ.2.OR.JEULER.EQ.1)) GO TO 35
      IF (IEULER(J).EQ.5 .AND. (JEULER.EQ.3.OR.JEULER.EQ.1)) GO TO 35
      IF (IEULER(J).EQ.4 .AND. (JEULER.EQ.3.OR.JEULER.EQ.2)) GO TO 35
      MODE = -1
      K = JEULER
      IF (K.GT.3) GO TO 33
      IF (K.EQ.2) GO TO 34
      K4 = 4-K
      CALL CROSS (HIR(1,K4,J),HIR(1,2,J),T)
      IEULER(J) = 8
      IPIN(J)   = 4
```

```
      CALL IMPLS2(MODE,J,T)
      I = -1
      GO TO 35
   33 MODE =  1
      K = K-3
      IF (K.GT.3) MODE=0
   34 IEULER(J) = 8
      IPIN(J)   = 4
      CALL IMPLS2(MODE,J,HIR(1,K,J))
      I = -1
   35 IEULER(J) = JEULER
      IPIN(J) = 4
      IF (IEULER(J).NE.8) IPIN(J) = -4
   36 CONTINUE
C
   37 IF  (NQ.LE.0)  GO TO 41
      DO 40 K=1,NQ
      IF (KQTYPE(K).LT.3) GO TO 40
      IF (KQTYPE(K).GT.4) GO TO 40
      IF (CFQQ(K).LT.0.0) KQTYPE(K) = -KQTYPE(K)
      IF (CFQQ(K).LT.0.0) GO TO 39
C
C     TEST IF ROLLING CONSTRAINT SHOULD BE SLIDING AND VICE VERSA.
C
      QN = -XDY(TQQ(1,K),HHT(1,1,K),QQ(1,K))
      IF (NPRT(24).NE.0) WRITE (6,38) KQTYPE(K),KQ1(K),KQ2(K),
     *                    (RK1(II,K),II=1,3),(RK2(II,K),II=1,3),
     *                    ((HHT(II,J,K),J=1,3),II=1,3),
     *       (QQ(II,K),II=1,3),(TQQ(II,K),II=1,3),(RQQ(II,K),II=1,3),
     *               (HQQ(II,K),II=1,3),SQQ(K),CFQQ(K),QN
   38 FORMAT('0 UPDATE ROLL-SLIDE TEST'/(2X,9G14.6))
      IF (QN.LT.0.0) KQTYPE(K) = -4
      IF (QN.LT.0.0) GO TO 39
      QDOTQ = QQ(1,K)**2 + QQ(2,K)**2 + QQ(3,K)**2
      QT = DSQRT(QDOTQ-QN**2)
      IF (KQTYPE(K).EQ.3 .AND. QT.LE.CFQQ(K)*QN) GO TO 40
      IF (KQTYPE(K).EQ.4 .AND. QT.GE.0.9*CFQQ(K)*QN) GO TO 40
      KQTYPE(K) = 7-KQTYPE(K)
   39 CALL OUTPUT(0)
      CALL SETUP2
      CALL DAUX(K)
      IF  (NPRT(24).NE.0)  CALL OUTPUT(1)
      IF  (NPRT( 3).NE.0)  CALL PRINT (6HUPDATE)
      I = -1
   40 CONTINUE
   41 CALL ELTIME(2,7)
   42 RETURN
      END
```

```
      SUBROUTINE VEHPOS
C                                                          REV 19 09/15/78
C     COMPUTES COMPONENTS OF VEHICLE ACCELERATIONS ONLY AS A FUNCTION
C     OF TIME USING DATA AND TABLES PRODUCED BY SUBROUTINE VINPUT.
C
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36)
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)
      COMMON/VPOSTN/ ZPLT(3),SPLT(3),AXV(3,6),VATAB(6,101,6),
     *               VTO(6),VDT(6),TIMEV(6),OMEGV(6),NVTAB(6),INDXV(6)
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),
     *               UNITL,UNITM,UNITT,GRAVTY(3)
      DIMENSION AX(3)
      T = TIME
      M = 1
   15 DO  16  I=1,3
   16 AX(I) = AXV(I,M)
      ATO = VTO(M)
      ADT = VDT(M)
      VTIME = TIMEV(M)
      OMEG  = OMEGV(M)
      NATAB = NVTAB(M)
      K = INDXV(M)
      IF(NATAB.NE.0) GO TO 20
C
C     HALF-SINE WAVE DECELERATION
C
      IF(T.GT.VTIME) T=VTIME
      WT = OMEG*T
      SWT = DSIN(WT)
      DO 10 I=1,3
      AW = AX(I)*OMEG
      SEGLA(I,K) = -AW*OMEG*SWT
   10 WMEGD(I,K) = 0.0
      GO TO 99
   20 IF (NATAB.LT.0) GO TO 30
C
C      UNIDIRECTIONAL DECELERATION
C
      IF (T.LT.VTIME) GO TO 21
C
C     TIME POINT EXCEEDS TABLE, USE LAST VALUES OF ACCELERATION.
C
      ACO = VATAB(1,NATAB,M)
      GO TO 25
C
C          USE QUADRATIC INTERPOLATION FROM TABLES FOR CURRENT VALUE OF
C          TIME TO BE CONSISTENT WITH SIMPSON INTEGRATION OF TABLES.
```

```
C
   21 J= 0.5*(T-ATO)/ADT +1.0
      XK = T/ADT -DFLOAT(2*J-1)
      X1 = XK+1.0
      X3 = XK-1.0
      ACO = 0.5*XK*X3*VATAB(1,2*J-1,M)
     *    -      X3*X1*VATAB(1,2*J   ,M)
     *    + 0.5*XK*X1*VATAB(1,2*J+1,M)
C
C        COMPONENTS OF VEHICLE ACCELERATION.
C
   25 DO 29 I=1,3
      SEGLA(I,K) = -G*AX(I)*ACO
   29 WMEGD(I,K) = 0.0
      GO TO 99
C
C      OMNIDIRECTIONAL DECELERATION
C
   30 J = (TIME-ATO)/ADT + 1.0
      IF (J.GE.-NATAB) GO TO 32
C
C          INTERPOLATION FROM VINPUT TABLES OF COMPONENTS OF VEHICLE
C          LINEAR AND ANGULAR ACCELERATION.
C
      TJ        = ATO + DFLOAT(J-1)*ADT
      DLT       = TIME-TJ
      R1 = DLT/ADT
      R2 = 1.0-R1
      DO 31 I=1,3
      SEGLA(I,K) =      -G*(VATAB(I  ,J+1,M)*R1 + VATAB(I  ,J,M)*R2)
   31 WMEGD(I,K) = RADIAN*(VATAB(I+3,J+1,M)*R1 + VATAB(I+3,J,M)*R2)
      GO TO 99
C
C     TIME POINT EXCEEDS TABLE, USE LAST VALUES OF ACCELERATION.
C
   32 J = - NATAB
      DO 33 I=1,3
      SEGLA(I,K) =      -G*VATAB(I  ,J,M)
   33 WMEGD(I,K) = RADIAN*VATAB(I+3,J,M)
   99 M = M+1
      IF  (M.LE.6 .AND. INDXV(M).NE.0)  GO TO 15
      RETURN
      END
```

```
      SUBROUTINE VINPUT
C                                                     REV 19 06/08/79
C     PERFORMS CARD INPUT AND COMPUTES DATA AND TABLES REQUIRED BY
C     SUBROUTINE VEHPOS TO INTEGRATE THE CRASH VEHICLE MOTION FOR ONE OF
C     THREE PERMISSABLE OPTIONS:
C       (1) HALF SINE-WAVE LINEAR DECELERATION IMPULSE
C       (2) UNIDIRECTIONAL LINEAR DECELERATION TABULAR INPUT
C       (3) OMNIDIRECTIONAL LINEAR AND ANGULAR ACCELERATION TABULAR
C            INPUT (6 DEGREES OF FREEDOM VEHICLE MOTION)
C
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36)
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)
      COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(3,60),HA(3,60),HB(3,60),
     *               RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90),
     *               JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30)
      COMMON/VPOSTN/ ZPLT(3),SPLT(3),AXV(3,6),VATAB(6,101,6),
     *               VT0(6),VDT(6),TIMEV(6),OMEGV(6),NVTAB(6),INDXV(6)
      COMMON/TEMPVS/ X0(3),XDOT0(3),XCOMP(3),XVCOMP(3),ANGLE(3),
     *               ATAB(15,100),DVEH(3,3),VMEG(3),VMEGD(3),
     *              XACOMP(3),THET(3),AX(3),F(5,100),XYZ(6,102),TT(102),
     *               VIPS,VMPH,AT0,ADT,VTIME,OMEG,NATAB
      COMMON/INTEST/ SGTEST(3,4,30),XTEST(3,120),SEGT(120),REGT(120)
      REAL    SEGT
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),
     *               UNITL,UNITM,UNITT,GRAVTY(3)
      COMMON/TITLES/ DATE(3),COMENT(40),VPSTTL(20),BDYTTL(5),
     *               BLTTTL(5,8),PLTTL(5,30),BAGTTL(5,6),SEG(30),
     *               JOINT(30),CGS(30),JS(30)
      REAL DATE,COMENT,VPSTTL,BDYTTL,BLTTTL,PLTTL,BAGTTL,SEG,JOINT
      LOGICAL*1 CGS,JS
      REAL VEH(6),GRND
      DATA VEH/4HVEH1,4HVEH2,4HVEH3,4HVEH4,4HVEH5,4HVEH /,GRND/4HGRND/
      DIMENSION IDYPR(3)
      DATA IDYPR/3,2,1/
C
C     READ AND PRINT CONTENTS OF CARDS C.1 AND C.2
C
      NVEH = NSEG
      NVH  = 0
      DO   11  I=1,6
   11 INDXV(I) = 0
   12 READ (5,13) VPSTTL
   13 FORMAT (20A4)
      READ(5,14) ANGLE,VIPS,VTIME,X0,NATAB,AT0,ADT,MSEG
   14 FORMAT(8F6.0,I6,2F6.0,I6)
      WRITE (6,15) VPSTTL,ANGLE,VIPS,VTIME,X0,NATAB,AT0,ADT,MSEG
   15 FORMAT('1 VEHICLE DECELERATION INPUTS',91X,'CARDS C'//3X,20A4//
```

```
     *     7X,'YAW',9X,'PITCH',7X,'ROLL',8X,'VIPS',8X,'VTIME',7X,'XO(X)',
     *     7X,'XO(Y)',7X,'XO(Z)',2X,'NATAB',6X,'ATO',9X,'ADT',4X,'MSEG'/
     *     8F12.3,I5,2X,2F12.6,I5)
      DA1 = ANGLE(1)*RADIAN
      DA2 = ANGLE(2)*RADIAN
      AX(3) = DCOS(DA2)
      AX(1) = DCOS(DA1)*AX(3)
      AX(2) = DSIN(DA1)*AX(3)
      AX(3) = DSIN(DA2)
      IF(NATAB.NE.0) GO TO 18
C
C     HALF-SINE WAVE DECELERATION
C
      OMEG = PI/VTIME
      AT = 0.5*VIPS/OMEG
      IF (VIPS.LT.0.0) VIPS = 0.0
      DO 16 I=1,3
      XACOMP(I) = 0.0
      XDOTO(I) = VIPS*AX(I)
   16 AX(I) = AT*AX(I)
      WRITE (6,17) VIPS,UNITL,UNITT,ANGLE,VTIME,UNITT
   17 FORMAT('0 PASSENGER COMPARTMENT DISPLACEMENT HISTORY'/
     *        '  ANALYTICAL HALF-SINE WAVE DECELERATION'/
     *        '  V0=',F8.3,1X,A4,'/',A4,', OBLIQUE ANGLES =',3F7.2,
     *        ' DEGREES, TIME DURATION =',F7.3,1X,A4//)
      GO TO 28
   18 IF (NATAB.LT.0) GO TO 31
C
C     FOR UNIDIRECTIONAL VEHICLE MOTION
C     READ LINEAR DECELERATION TABLES FROM CARDS C.3
C
      READ (5,19) (ATAB(1,I),I=1,NATAB)
   19 FORMAT (12F6.0)
C
C     EXTEND TABLE IF NECESSARY SUCH THAT NATAB IS ODD AND
C     LAST ENTRY NEED NOT BE ZERO. IF TABLE SIZE IS EXCEEDED ON TIME,
C     VALUE OF LAST ENTRY WILL BE USED.
C
      IF (MOD(NATAB,2).EQ.1) GO TO 20
      ATAB(1,NATAB+1) = ATAB(1,NATAB)
      NATAB = NATAB+1
   20 VTIME = ADT * DFLOAT(NATAB-1)
C
C     USING SIMPSON'S INTEGRATION, COMPUTE VELOCITY AND DISPLACEMENT
C     TABLE FOR NATAB EQUALLY SPACED (ADT) TIME POINTS.
C     FOR I=1,NATAB
C       ATAB(1,I) = LINEAR DECELERATION (G'S)
C       ATAB(2,I) = LINEAR VELOCITY (L UNITS/T UNITS)
C       ATAB(3,I) = LINEAR DISPLACEMENT (L UNITS)
C
```

```
      ATAB(2,1) = VIPS
      ATAB(3,1) = 0.0
      DA1 = ADT/3.0
      DA2 = ADT/12.0
      UNITS = -G
      DO 22 J=2,3
      DO 21 I=2,NATAB,2
      F1 = ATAB(J-1,I-1) * UNITS
      F2 = ATAB(J-1,I  ) * UNITS
      F3 = ATAB(J-1,I+1) * UNITS
      ATAB(J,I  ) = ATAB(J,I-1) + DA2*(5.0*F1+8.0*F2-F3)
   21 ATAB(J,I+1) = ATAB(J,I-1) + DA1*(    F1+4.0*F2+F3)
   22 UNITS = 1.0
C
C     PRINT TABLES
C
      WRITE (6,23) (UNITL,UNITT,UNITL,I=1,2)
   23 FORMAT('0 UNIDIRECTIONAL VEHICLE POSITION TABLES'//
     *     2('      TIME        ACC        VELOCITY      POSITION    ')/
     *   2('     (MSEC)      (G)     (',A4,'/',A4,')',5X,'(',A4,')',4X)/)
      DO 26 J=1,50
      IF (J.GT.NATAB) GO TO 26
      T1 = (AT0 + DFLOAT(J-1)*ADT)*1000.0
      IF (J+50.LE.NATAB) GO TO 25
      WRITE (6,24) T1,(ATAB(I,J),I=1,3)
   24 FORMAT(2(F11.5,F10.2,F13.4,F13.5,3X))
      GO TO 26
   25 T2 = (AT0 + DFLOAT(J+49)*ADT)*1000.0
      WRITE (6,24) T1,(ATAB(I,J),I=1,3),T2,(ATAB(I,J+50),I=1,3)
   26 CONTINUE
C
C     INITIALIZATION
C
      DO 27 I=1,3
      XACOMP(I) = -G*AX(I)*ATAB(1,1)
   27 XDOT0(I)= VIPS*AX(I)
   28 DO 30 I=1,3
      DO 29 J=1,3
   29 DVEH(I,J) = 0.0
      DVEH(I,I) = 1.0
      VMEGD(I) = 0.0
   30 VMEG(I)   = 0.0
      GO TO 64
C
C     FOR OMNIDIRECTIONAL (6 DEGREES OF FREEDOM) VEHICLE MOTION
C     READ LINEAR DECELERATION AND ANGULAR ACCELERATION TABLES
C     FROM CARDS C.4.
C
   31 MATAB = -NATAB
      READ  (5,32)  LTYPE,LFIT,NPTS,(VMEG(I),I=1,3)
```

```
32 FORMAT (3I6,22X,3F10.0)                                            VINPUT
   IF  (LTYPE.GT.0)  GO TO 34                                         VINPUT
   READ (5,33) ((ATAB(I,J),I=1,3),(ATAB(I,J),I=10,12),J=1,MATAB)      VINPUT
33 FORMAT (10X,6F10.0)                                                VINPUT
   ISKIP = 0                                                          VINPUT
   GO TO 46                                                           VINPUT
34 LPTS = LTYPE-1 + NPTS                                              VINPUT
   READ  (5,35)  (TT(I),(XYZ(I,J),J=1,6),I=1,LPTS)                    VINPUT
35 FORMAT (7F10.0)                                                    VINPUT
   WRITE  (6,36)  LTYPE,LFIT,NPTS,                                    VINPUT
  *               (TT(I),(XYZ(I,J),J=1,6),I=1,LPTS)                   VINPUT
36 FORMAT ('0 SPLINE FIT TABULAR INPUT'//                             VINPUT
  *        3X,'LTYPE =',I6,'  LFIT =',I6,'  NPTS =',I6//              VINPUT
  *        (F15.6,3X,3F12.3,3X,3F12.3))                               VINPUT
   DO  37  I=1,3                                                      VINPUT
   I4 = 4-I                                                           VINPUT
   X0(I) = XYZ(1,I)                                                   VINPUT
   IF  (LTYPE.EQ.1)  GO TO 37                                         VINPUT
   XDOTO(I) = XYZ(2,I)                                                VINPUT
   VMEG(I) = XYZ(2,I+3)                                               VINPUT
37 ANGLE(I4) = XYZ(1,I+3)                                             VINPUT
   DO  45  II=1,6                                                     VINPUT
   CALL SPLINE (TT(LTYPE),XYZ(LTYPE,II),F,NPTS,LFIT)                  VINPUT
   I = II                                                             VINPUT
   IF  (II.GT.3)  I = II + 6                                          VINPUT
   IF  (LTYPE.NE.1)  GO TO 38                                         VINPUT
   IF  (II.LE.3)  XDOTO(I) = F(3,1)                                   VINPUT
   IF  (II.GT.3)  VMEG(II-3) = F(3,1)                                 VINPUT
   IF  (II.GT.3)  I = 16-II                                           VINPUT
38 UNITS = 1.0                                                        VINPUT
   IF  (LTYPE.LT.3 .AND. II.LE.3)  UNITS = -1.0/G                     VINPUT
   K1 = 1                                                             VINPUT
   DO  45  J=1,MATAB                                                  VINPUT
   TTT = AT0 + DFLOAT(J-1)*ADT                                        VINPUT
   DO  39  K=K1,NPTS                                                  VINPUT
   IF  (K.EQ.NPTS)  GO TO 40                                          VINPUT
   IF  (DABS(TTT-F(1,K+1)).LT.EPS(8))  TTT = F(1,K+1)                 VINPUT
   IF  (TTT.LT.F(1,K+1))  GO TO 40                                    VINPUT
39 CONTINUE                                                           VINPUT
40 K1 = K                                                             VINPUT
   DX = TTT - F(1,K)                                                  VINPUT
   GO TO (41,42,43),LTYPE                                             VINPUT
41 ACC = 2.0*F(4,K) + 6.0*DX*F(5,K)                                   VINPUT
   GO TO 44                                                           VINPUT
42 ACC = F(3,K) + DX*(2.0*F(4,K)+3.0*DX*F(5,K))                       VINPUT
   GO TO 44                                                           VINPUT
43 ACC = F(2,K) + DX*(F(3,K)+DX*(F(4,K)+DX*F(5,K)))                   VINPUT
44 ATAB(I,J) = ACC*UNITS                                              VINPUT
45 CONTINUE                                                           VINPUT
   ISKIP = 1                                                          VINPUT
```

```
   46 DO  55  J=1,MATAB                                                  VINPUT
      IF (MOD(J,45).NE.1) GO TO 49                                       VINPUT
C                                                                        VINPUT
C     PRINT PAGE HEADING AT START OF EACH 45 TIME POINTS.                VINPUT
C                                                                        VINPUT
      IPAGE = (J-1)/45 + 1                                               VINPUT
      WRITE (6,48) ISKIP,VPSTTL,IPAGE,UNITL,UNITT,UNITL                  VINPUT
   48 FORMAT(I1,'  VEHICLE LINEAR TIME HISTORY',3X,20A4,3X,              VINPUT
     *          'PAGE NO.',I3//                                          VINPUT
     *       4X,'TIME',12X,'LINEAR DECELERATIONS (G''S)',                VINPUT
     *                 11X,'LINEAR VELOCITIES (',A4,'/',A4,')',          VINPUT
     *                 11X,'LINEAR DISPLACEMENTS (',A4,')'     /         VINPUT
     *        3X,'(MSEC)',3(11X,'X',11X,'Y',11X,'Z',3X)      /  )        VINPUT
      ISKIP = 1                                                          VINPUT
   49 IF (J.GT.1) GO TO 52                                               VINPUT
C                                                                        VINPUT
C     INTEGRATION INITIALIZATION FOR TIME = 0.                           VINPUT
C                                                                        VINPUT
      DO 50 I=1,3                                                        VINPUT
      ATAB(I+6,J) = X0(I)                                                VINPUT
      ATAB(I+12,J) = VMEG(I)                                             VINPUT
   50 THET(I) = ANGLE(I)*RADIAN                                          VINPUT
      CALL DRCYPR (DVEH,ANGLE,IDYPR)                                     VINPUT
      DO 51 I=1,3                                                        VINPUT
      IF (LTYPE.EQ.0)  XDOTO(I) = VIPS*DVEH(1,I)                         VINPUT
   51 ATAB(I+3,J) = XDOTO(I)                                             VINPUT
      GO TO 54                                                           VINPUT
   52 DO 53 I=1,3                                                        VINPUT
C                                                                        VINPUT
C     INTEGRATE LINEAR VELOCITY AND DISPLACEMENT.                        VINPUT
C                                                                        VINPUT
      ATAB(I+3,J) = ATAB(I+3,J-1)-G*ADT/2.0*(ATAB(I,J-1)+ATAB(I,J))      VINPUT
   53 ATAB(I+6,J) = ATAB(I+6,J-1)                                        VINPUT
     *    +ADT*(ATAB(I+3,J-1)-G*ADT/6.0*(2.0*ATAB(I,J-1)+ATAB(I,J)))     VINPUT
   54 T1 = (AT0 + DFLOAT(J-1)*ADT)*1000.0                                VINPUT
   55 WRITE(6,56) T1,(ATAB(I,J),I=1,9)                                   VINPUT
   56 FORMAT(F9.3,3(3X,3F12.3))                                          VINPUT
      DO 61 J=1,MATAB                                                    VINPUT
      IF (MOD(J,45).NE.1) GO TO 58                                       VINPUT
C                                                                        VINPUT
C     PRINT PAGE HEADING AT START OF EACH 45 TIME POINTS.                VINPUT
C                                                                        VINPUT
      IPAGE = (J-1)/45 + 1                                               VINPUT
      WRITE (6,57) VPSTTL,IPAGE,UNITT,UNITT                              VINPUT
   57 FORMAT('1 VEHICLE ANGULAR TIME HISTORY',3X,20A4,3X,'PAGE NO.',I3// VINPUT
     *        4X,'TIME', 7X,'ANGULAR ACCELERATIONS (DEG/',A4,'**2)',     VINPUT
     *                   7X,'ANGULAR VELOCITIES (DEG/',A4,')',           VINPUT
     *                  11X,'ANGULAR DISPLACEMENTS (DEG)' /              VINPUT
     *       3X,'(MSEC)',2(11X,'X',11X,'Y',11X,'Z',3X),                  VINPUT
     *       10X,'YAW',8X,'PITCH',8X,'ROLL' /)                           VINPUT
```

```
   58 IF(J.EQ.1) GO TO 60
C
C     INTEGRATE ANGULAR VELOCITY AND DISPLACEMENT.
C
      DO 59 I=1,3
      ATAB(I+12,J) = ATAB(I+12,J-1)+(ATAB(I+9,J-1)+ATAB(I+9,J))*ADT/2.0
   59 THET(I) = ADT*(ATAB(I+12,J-1)+(2.0*ATAB(I+9,J-1)+ATAB(I+9,J))*ADT
     */6.0)*RADIAN
      CALL DSETD(DVEH,THET,THT)
   60 CALL YPRDEG(DVEH,THET)
      T1 = (ATO + DFLOAT(J-1)*ADT)*1000.0
   61 WRITE (6,56)  T1,(ATAB(I,J),I=10,15),THET
C
C     PROGRAM INITIALIZATION FOR TIME = 0.
C
      CALL DRCYPR (DVEH,ANGLE,IDYPR)
      DO 63 I=1,3
      XACOMP(I) = -G*ATAB(I,1)
      VMEG(I)  = ATAB(I+12,1)*RADIAN
   63 VMEGD(I) = ATAB(I+9 ,1)*RADIAN
   64 J = MSEG
      IF  (MSEG.EQ.0)  GO TO 65
      IF  (MSEG.LE.NSEG)  GO TO 66
      IF  (MSEG.NE.NVEH+1)  STOP 6
   65 NVEH = NVEH+1
      J = NVEH
C
C     SETUP FOR ALL PRESCRIBED SEGMENT MOTION.
C
   66 NVH = NVH+1
      ISING(J) = -1
      IF  (MSEG.GT.NSEG)  SEG(J) = VEH(NVH)
      DO  67  I=1,3
      SEGLA(I,J) = VMEGD(I)
      WMEGD(I,J) = XACOMP(I)
   67 AXV(I,NVH) = AX(I)
      VTO(NVH) = ATO
      VDT(NVH) = ADT
      OMEGV(NVH) = OMEG
      TIMEV(NVH) = VTIME
      NVTAB(NVH) = NATAB
      INDXV(NVH) = J
      NJ = IABS(NATAB)
      IF  (NJ.LE.0)  GO TO 69
      DO  68  K=1,NJ
      DO  68  I=1,3
      VATAB(I  ,K,NVH) = ATAB(I,K)
   68 VATAB(I+3,K,NVH) = ATAB(I+9,K)
   69 IF  (J.LE.NSEG)  GO TO 72
C
```

```
C     SETUP FOR NEW VEHICLE (SEGMENT) MOTION.
C
      W(J)  = 0.0
      RW(J) = 0.0
      DO  71  I=1,3
      DO  70  K=1,3
      D(I,K,J) = DVEH(I,K)
   70 SGTEST(I,K,J) = 0.0
      SGTEST(I,4,J) = 0.0
      SEGLP(I,J) = XO(I)
      SEGLV(I,J) = XDOTO(I)
      WMEG (I,J) = VMEG(I)
      PHI  (I,J) = 0.0
   71 RPHI (I,J) = 0.0
   72 IF  (MSEG.NE.0)  GO TO 12
      SEG(NVEH) = VEH(6)
C
C     SET UP SEGMENT DATA FOR GROUND
C
      NGRND = NVEH+1
      IF  (NGRND.GT.30 .OR. NVH.GT.6)  STOP 7
      SEG(NGRND) = GRND
      J = NGRND
      ISING(J) = -1
      W(J)  = 0.0
      RW(J) = 0.0
      DO  74  I=1,3
      DO  73  K=1,3
      D(I,K,J) = 0.0
   73 SGTEST(I,K,J) = 0.0
      D(I,I,J) = 1.0
      SGTEST(I,4,J) = 0.0
      SEGLP(I,J) = 0.0
      SEGLV(I,J) = 0.0
      SEGLA(I,J) = 0.0
      WMEG (I,J) = 0.0
      WMEGD(I,J) = 0.0
      PHI  (I,J) = 0.0
   74 RPHI (I,J) = 0.0
      RETURN
      END
```

```
      SUBROUTINE VISPR(IJ,NJ)
C                                                     REV 19 10/30/79
C     COMPUTES VISCOS AND SPRING TORQUES AT THE JOINTS
C     AND ADDS THEM TO THE U2 ARRAY.
C
C     ARGUMENTS:
C        NJ = 0 - REGULAR COMPUTATION FOR ALL JOINTS
C           # 0 - COMPUTE ONLY FOR JOINT NJ IMPULSE
C
C        IJ = 1 IMPULSE FOR FLEXURE ONLY
C           = 2 IMPULSE FOR TORSION ONLY
C           = 4 IMPULSE FOR GLOBALGRAPHIC ONLY
C
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36)
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)
      COMMON/DESCRP/ PHI(3,30),W(30),RW(30),SR(3,60),HA(3,60),HB(3,60),
     *               RPHI(3,30),HT(3,3,60),SPRING(5,90),VISC(7,90),
     *               JNT(30),IPIN(30),ISING(30),IGLOB(30),JOINTF(30)
      COMMON/CMATRX/ V1(3,30),V2(3,30),V3(3,12),B12(3,3,60),A22(3,3,60),
     *               F(3,30),TQ(3,30),WJ(30)
      COMMON/FORCES/ PSF(7,30),BSF(4,20),SSF(10,20),BAGSF(3,20),
     *               PRJNT(6,30),NPANEL(5),NPSF,NBSF,NSSF,NBGSF
      COMMON/CEULER/ IEULER(30),HIR(3,3,30),ANG(3,30),ANGD(3,30),
     *               FE(3,30),TQE(3,30),CONST(3,30)
      COMMON/TEMPVI/ CREST,TTI(3),R1I(3),R2I(3),JSTOP(4,2,30)
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),
     *               UNITL,UNITM,UNITT,GRAVTY(3)
      COMMON/TEMPVS/ T3(3),T6(3),T7(3),T8(3),T9(3),
     *               WIJ(3),ANGL(3),DH1(3,3),HD3(3,3),
     *               HAD,HBD,WIJM,CV,CSA,CSB,TQC
      IF (NJNT.LE.0) GO TO 99
      CALL ELTIME(1,13)
      IF (NPRT(12).NE.0) WRITE (6,11)  TIME
   11 FORMAT('1 VISPR COMPUTATIONS FOR TIME =',F12.6)
      J1 = 1
      J2 = NJNT
      IF (NJ.EQ.0) GO TO 13
      J1 = NJ
      J2 = NJ
   13 DO 90 J=J1,J2
      DO 12 L=1,3
      T3(L) = 0.0
      T6(L) = 0.0
      ANGL(L) = 0.0
   12 TQ(L,J) = 0.0
      WJ(J) = 0.0
C
```

```
C     DO NOT COMPUTE TORQUES FOR NULL, LOCKED OR EULER JOINTS.
C
      I = IABS(JNT(J))
      IF (I.LE.0) GO TO 90
      CALL DOT33 (D(1,1,J+1),HT(1,1,2*J),HIR(1,1,J))
      IF (IPIN(J).LT.0 .OR. IPIN(J).GT.3) GO TO 90
C
C     ZERO T1-T9 ARRAYS AND HAD,HBD,WIJM,CV,CS4,CSB AND TQC.
C
      WIJM = 0.0
      CV   = 0.0
      CSA  = 0.0
      CSB  = 0.0
      TQC  = 0.0
      CALL DOT33 (D(1,1,I),HT(1,1,2*J-1),DH1)
      CALL DOT33 (DH1,HIR(1,1,J),HD3)
      HAD  = HD3(3,3)
      IF  (HAD.GT. 1.0)  HAD =  1.0
      IF  (HAD.LT.-1.0)  HAD = -1.0
      ANGL(1) = DARCOS(HAD)
      IF  (HD3(2,3).NE.0.0 .OR. HD3(1,3).NE.0.0)
     *ANGL(2) = DATAN2(HD3(2,3),HD3(1,3))
      ANGL(3) = DATAN2(HD3(2,1)-HD3(1,2),HD3(1,1)+HD3(2,2))
      IF (N).NE.0.AND.IJ.EQ.4) GO TO 27
C
C     CONVERT TO INERTIAL REFERENCE SYSTEM
C        T1= D(I)'*HA(NJ)      T4=D(J+1)'*HA(MJ)
C        T3= D(I)'*WMEG(I)     T6=D(J+1)'*WMEG(J+1)
C
C     HAD = COS TA = T1.T4
C     WIJ = T3-T6
C     WJ  = |WIJ|
C
      DO 20 L=1,3
      DO 15 M=1,3
      T3(L) = T3(L)+ D(M,L,I)* WMEG(M,I)
   15 T6(L) = T6(L)+ D(M,L,J+1)* WMEG(M,J+1)
      WIJ(L)= T3(L)-T6(L)
   20 WIJM = WIJM + WIJ(L)**2
      WIJM = DSQRT(WIJM)
      WJ(J) = WIJM
C
C     T7 = T1 X T4
C     HAC = |T7|
C
      CALL CROSS (DH1(1,3),HIR(1,3,J),T7)
      HACC = T7(1)**2 + T7(2)**2 + T7(3)**2
      HAC = DSQRT(HACC)
C
C     COMPUTE CV, THE MAGNITUDE OF VISCOUS AND COULOMB TORQUE/WIJM
```

```
C            RA = +SGN TA DOT = -WIJ.T7
C         AND CSA. THE MAGNITUDE OF FLEXURE TORQUE/HAC
C
      CV = VISCOS(WIJM,VISC(1,3*J-2),HA2)
      IF  (NJ.EQ.0)  HA(2,2*J) = HA2
      CREST = VISC(7,3*J-2)
      RA = -(WIJ(1)*T7(1) + WIJ(2)*T7(2) + WIJ(3)*T7(3))
      IF  (HAC.NE.0.0)  RA = RA/HAC
      JSTP = 0
      IF (JOINTF(J).EQ.0) CSA = EFUNCT(ANGL(1),RA,SPRING(1,3*J-2),JSTP)
      IF (JOINTF(J).NE.0) CSA = FNTERP(ANGL(1),ANGL(2),JOINTF(J))
      IF  (HAC.NE.0.0)  CSA = CSA/HAC
      IF (NJ.EQ.0) JSTOP(1,1,J) = JSTP
      IF (IPIN(J).EQ.1) GO TO 34
C
C     RB = +SGN TB DOT = -WIJ.T8
C     COMPUTE CSB, THE MAGNITUDE OF TORSIONAL TORQUE/HBC
C
      RB  = -(WIJ(1)*HIR(1,3,J) + WIJ(2)*HIR(2,3,J) + WIJ(3)*HIR(3,3,J))
      CSB = EFUNCT(ANGL(3),RB,SPRING(1,3*J-1),JSTP)
      IF (NJ.EQ.0) JSTOP(2,1,J) = JSTP
      IF (NJ.GT.0) GO TO 34
C
C     COMPUTE EFFECT OF GLOBALGRAPHIC JOINT STOP (IPIN=3)
C
   27 IF (IPIN(J).NE.3) GO TO 34
      CALL GLOBAL (J,HD3(1,3),DH1,TQC,T9,ANGL)
C
C     COMPUTE TOTAL TORQUE IN INERTIAL REFERENCE BY
C     TQ = -CV*WIJ +.CSA*T7 + CSB*T8 + TQC*T9
C
   34 IF (NJ.EQ.0) GO TO 36
      CV = 0.0
      IF (IJ.NE.1) CSA = 0.0
      IF (IJ.NE.2) CSB = 0.0
      IF (IJ.NE.4) TQC = 0.0
      IF (HA(2,2*J).EQ.0.0) GO TO 36
      CALL MAT31 (HIR(1,1,J),HA(1,2*J-1),TQ(1,J))
      DO 38 L=1,3
   38 TQ(L,J) = HA(2,2*J)*TQ(L,J)
   36 DO 37 L=1,3
      TQ(L,J) = TQ(L,J) -CV*WIJ(L) +CSA*T7(L) +CSB*HIR(L,3,J) +TQC*T9(L)
   37 TTI(L) = TQ(L,J)
      IF (NPRT(12).NE.0) WRITE (6,39)
     *          J,CV,CSA,CSB,HAC,RA,RB,(TQ(L,J),L=1,3),
     *          WIJ,T7,ANGL, DH1 , HD3,
     *          ((HIR(L,K,J),L=1,3),K=1,3)
   39 FORMAT(I4,1P9D14.6/(4X,9D14.6))
C
C     ADD TORQUE CONVERTED TO LOCAL REFERENCE BY
```

```
C       U2I = U2I + DI*TQ
C       U2J = U2J - DJ*TQ
C
      DO 40 L=1,3
      DO 40 M=1,3
      U2(L,I  ) = U2(L,I  ) + D(L,M,I  )*TQ(M,J)
   40 U2(L,J+1) = U2(L,J+1) - D(L,M,J+1)*TQ(M,J)
C
C     STORE DATA FOR OUTPUT ROUTINE INTO PRJNT ARRAY.
C
      PRJNT(1,J) = ANGL(1)
      PRJNT(2,J) = ANGL(3)
      PRJNT(3,J) = CSA*HAC
      PRJNT(4,J) = CSB
      PRJNT(5,J) = CV*WIJM
      PRJNT(6,J) = DSQRT(TQ(1,J)**2+TQ(2,J)**2+TQ(3,J)**2)
   90 CONTINUE
      CALL ELTIME(2,13)
   99 RETURN
      END
```

```
      SUBROUTINE WINDY(M,MM,N,NN,NT)
C                                                    REV 19 08/05/78
C     COMPUTES FORCES AND TORQUES ADDING THEM TO THE U1 AND U2 ARRAYS
C     OF WIND BLAST FORCES DETERMINED BY FUNCTION STORED IN TAB(NT)
C     ON ELLIPSOID (MM) ATTACHED TO BODY SEGMENT (M) WHICH EXTENDS
C     THROUGH THE INTERSECTING PLANE (NN) ATTACHED TO SEGMENT (N).
C
       IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/CONTRL/ TIME,NSEG,NJNT,NPL,NBLT,NBAG,NVEH,NGRND,
     *               NS,NQ,NSD,NFLX,NHRNSS,NWINDF,NJNTF,NPRT(36)
      COMMON/SGMNTS/ D(3,3,30),WMEG(3,30),WMEGD(3,30),U1(3,30),U2(3,30),
     *               SEGLP(3,30),SEGLV(3,30),SEGLA(3,30),NSYM(30)
      COMMON/TABLES/MXNTI,MXNTB,MXTB1,MXTB2,NTI(50),NTAB(500),TAB(2000)
      COMMON/CNTSRF/ PL(17,30),BELT(20,8),TPTS(6,8),BD(24,40)
      COMMON/CNSNTS/ PI,RADIAN,G,THIRD,EPS(24),
     *               UNITL,UNITM,UNITT,GRAVTY(3)
      COMMON/TEMPVS/ DMNT(3,3),XMN(3),XMM(3),TM(3),BET,BTS,P,FT(3),
     *               FF(3),AF(3),FAF,TF,BREF,SCALE,TRACER,AREA,RLM(3),
     *               TQM(3),RM(3)
      COMMON/WINDFR/ WTIME(30),IWIND(30),MWSEG(5,30)
      CALL ELTIME(1,37)
C
C     COMPUTE PENETRATION DISTANCE; IF NEGATIVE, RETURN.
C
      CALL DOTT33 (D(1,1,M),D(1,1,N),DMNT)
      DO 10 I=1,3
   10 XMN(I) = SEGLP(I,M) - SEGLP(I,N)
      CALL MAT31 (D(1,1,M),XMN,XMM)
      CALL MAT31 (DMNT,PL(1,NN),TM)
      BET = PL(4,NN)
      DO 11 I=1,3
   11 BET = BET - TM(I)*(BD(I+3,MM)+XMM(I))
      CALL MAT31 (BD(16,MM),TM,RM)
      BTS = TM(1)*RM(1) + TM(2)*RM(2) + TM(3)*RM(3)
      BTE = -DSQRT(BTS)
      P   = BET - BTE
      IF (P.LT.0.0) GO TO 99
C
C     FETCH OR STORE INITIAL PENETRATION TIME.
C
      IWIND(M) = M
      IF (TIME.LE.WTIME(M)) WTIME(M) = TIME
      FTIME = TIME - WTIME(M)
C
C     GET FORCE VECTOR FT FROM TABLE NT FOR TIME = FTIME.
C
   22 KT = NTI(NT)
      NENTRY = TAB(KT+5)
      K1 = KT+10
      K2 = 4*NENTRY + KT+2
```

```
      IF (NENTRY.EQ.1) GO TO 31
      DO 30 K=K1,K2,4
      IF (FTIME.GT.TAB(K)) GO TO 30
      KK = K
      R1 = (TAB(K)-FTIME)/(TAB(K)-TAB(K-4))
      GO TO 32
   30 CONTINUE
   31 KK = K2
      R1 = 0.0
   32 R2 = 1.0 - R1
      DO 33 I=1,3
      K= KK+I
   33 FT(I) = R2*TAB(K) + R1*TAB(K-4)
C
C     COMPUTE PRESENTED AREA TO WIND FORCE.
C
      CALL MAT31 (D(1,1,M),FT,FF)
      CALL MAT31 (BD(7,MM),FF,AF)
      FAF = FF(1)*AF(1) + FF(2)*AF(2) + FF(3)*AF(3)
      IF (FAF.LE.0.0) GO TO 99
      TF  = TM(1)*FF(1) + TM(2)*FF(2) + TM(3)*FF(3)
      BREF = DSQRT(BTS-TF*TF/FAF)
      SCALE = (-BET+BREF)/(-BTE+BREF)
      IF (SCALE.GE.1.0) GO TO 99
      IF (SCALE.LT.0.0) SCALE = 0.0
      TRACER = (BD( 7,MM)-AF(1)**2/FAF)*(BD(11,MM)-AF(2)**2/FAF)
     *         + (BD( 7,MM)-AF(1)**2/FAF)*(BD(15,MM)-AF(3)**2/FAF)
     *         + (BD(11,MM)-AF(2)**2/FAF)*(BD(15,MM)-AF(3)**2/FAF)
     *         - (BD( 8,MM)-AF(1)*AF(2)/FAF)**2
     *         - (BD( 9,MM)-AF(1)*AF(3)/FAF)**2
     *         - (BD(12,MM)-AF(2)*AF(3)/FAF)**2
      AREA = (1.0-SCALE**2) * PI / DSQRT(TRACER)
C
C     ADD FORCE AND TORQUES TO U1 AND U2 ARRAYS FOR SEGMENT M.
C
      SCALE = SCALE/BTE
      DO 36 I=1,3
      RLM(I) = RM(I)*SCALE + BD(I+3,MM)
      FT (I) = FT(I)*AREA
   36 FF (I) = FF(I)*AREA
      CALL CROSS (RLM,FF,TQM)
      DO 39 I=1,3
      U1(I,M) = U1(I,M) + FT(I)
   39 U2(I,M) = U2(I,M) + TQM(I)
      IF (NPRT(14).NE.0) WRITE (6,41) TIME,M,P,AREA,FT,TQM
   41 FORMAT(' WIND FORCE',F14.6,I6,2F10.3,3X,3F12.5,3X,3F12.5)
   99 CALL ELTIME (2,37)
      RETURN
      END
```

REFERENCES

Bartz, J.A., "A Three-Dimensional Computer Simulation of a Motor Vehicle Crash Victim, Phase I - Development of the Computer Program," Calspan Technical Report No. VJ-2978-V-1, July 1971.

Bartz, J.A. and Butler, F.E., "A Three-Dimensional Computer Simulation of a Motor Vehicle Crash Victim, Phase II - Validation of the Model," Calspan Technical Report No. VJ-2978-V-2, December 1972.

Fleck, J.T., Butler, F.E. and Vogel, S.L., "An Improved Three-Dimensional Computer Simulation of Crash Victims," NHTSA Report Nos. DOT-HS-801-507 through 510, National Technical Information Service, Springfield, Virginia 22161, Accession Nos. PB-241692, 3, 4 and 5, April 1975. (Formerly Calspan Report No. ZQ-5180-L-1, July 1974).

Volume I: Engineering Manual
Volume II: Model Validation
Volume III: User's Manual
Volume IV: Programmer's Manual

Fleck, J.T. and Butler, F.E., "Development of an Improved Computer Model of the Human Body and Extremity Dynamics," Report No. AMRL-TR-75-14, July 1975 (AD A-014816).

Strieb, M., "Structural Modification of the 'Three Dimensional Crash Victim Simulation' (3DCVS) Software," Technical Report No. 1194-A, Analytics Inc., Willow Grove, PA 19090, May 1976 (AD-A027726).

☆U.S.Government Printing Office: 1980 — 657-084/07

FILMED

9-8